J Heuristics
DOI 10.1007/s10732-014-9246-1

A distributed social choice protocol for combinatorial domains

Minyi Li · Quoc Bao Vo · Ryszard Kowalczyk

Received: 2 April 2013 / Revised: 24 November 2013 / Accepted: 19 March 2014
© Springer Science+Business Media New York 2014

Abstract In this paper, we study the problem of collective decision-making over combinatorial domains, where the set of possible alternatives is a Cartesian product of (finite) domain values for each of a given set of variables, and these variables are *not* preferentially independent. Due to the large alternative space, most common rules for social choice cannot be directly applied to compute a winner. In this paper, we introduce a distributed protocol for collective decision-making in combinatorial domains, which enjoys the following desirable properties: (i) the final decision chosen is guaranteed to be a Smith member; (ii) it enables distributed decision-making and works under incomplete information settings, i.e., the agents are not required to reveal their preferences explicitly; (iii) it significantly reduces the amount of dominance testings (individual outcome comparisons) that each agent needs to conduct, as well as the number of pairwise comparisons; (iv) it is sufficiently general and does not restrict the choice of preference representation languages.

Keywords Negotiation · Combinatorial domains · Structural preferences · CP-nets · Best possible agreement

1 Introduction

In classical social choice theory, candidates (aka. alternatives, outcomes) and the agents' preferences are supposed to be listed explicitly, which relies on a demanding assumption that the candidates should not be too numerous. In this paper, we

M. Li (✉) · Q. B. Vo · R. Kowalczyk
Swinburne University of Technology, Hawthorn, VIC, Australia
e-mail: liminyi0709@gmail.com; myli@swin.edu.au

Q. B. Vo
e-mail: bvo@swin.edu.au

R. Kowalczyk
e-mail: rkowalczyk@swin.edu.au

Published online: 05 April 2014

focus on the problem of collective decision-making in combinatorial domains, where the set of candidates is a Cartesian product of finite value domains for each of a given set of possibly inter-dependent variables. For instance, a research group plan to order several personal computers (PCs) and the group members need to decide on a common group PC configuration. Consider for example that the common PC configuration to be agreed on is composed of the processor, hard disk, RAM and operating system, with a choice of six possible options for each, which makes $6^4 = 1296$ candidates. The agents' preferences on variables and the collective decisions are not independent, because, perhaps, the preferred operating systems may depend on the given processor type. The inter-dependency between variables and the exponential alternative space in such domains lead to two important problems in this area: *how to express preferences* and *how to compute social choice rules*, since neither explicit listing of preferences, nor direct election (counting alternatives) for computing a "good" collective decision is applicable. The first problem of preference representation in combinatorial domains has been well studied in the literature, and several compact languages have been developed by the KR community (see e.g., Bacchus et al. 1995; Boutilier et al. 2004; Lang 2002). Therefore, we focus on the latter problem in this paper. It is worth noticing that in this work, we allow flexible usage of preference representation languages in combinatorial domains, i.e., there are no restrictions on the representation model used.

When the domain is combinatorial, a straightforward way to compute a social decision rule might be using issue-by-issue (a.k.a. seat-by-seat) sequential election. However, as soon as the variables are not preferentially independent, it is very likely that deciding on the issues separately will lead to suboptimal choices (Brams et al. 1998). The only way to avoid inefficiency is "deciding on combinations [of variable values]" (Brams et al. 1998). Nonetheless, most common aggregation methods need a number of operations at least linear (sometimes even quadratic or exponential) in the number of possible alternatives (Lang 2004). When the set of alternatives has a combinatorial structure and the agents' preferences are described in a compact representation language, generating the whole relation from those inputs and directly applying such rules to compute a collective decision are impractical.

Collective decision-making in combinatorial domains has been studied in a number of papers. Many of them consider logical or structured preferences, and mainly focus on analyzing negative complexity results or finding reasonable restrictions on the preference relations such that sequential winners and direct winners always coincide (e.g., Lang and Xia 2009; Xia et al. 2007). Some later works by Xia et al. (2008) and Li et al. (2011b) relax those restrictions and introduce the notion of *local Condorcet winners*. [1] They also propose (and implement) an algorithm for computing them. Nevertheless, with a limited number of agents and a large alternative space, collective preference cycles (i.e., paradoxes) generally occur, where there may not exist a clear winner. Therefore, investigating techniques to generate a "good" alternative in the presence of paradoxes is certainly of interest in a number of circumstances.

In this paper, we address the aforementioned problem by proposing a protocol that enables the agents to identify efficient collective decisions in combinatorial domains.

[1] A local Condorcet winner is an alternative **x** that beats all its neighbors. And a neighbor of **x** is another alternative that differs only on a single variable from **x**.

Given a social choice rule, the proposed protocol enjoys the following desirable properties: (i) Regarding the social properties, the final chosen collective decision is guaranteed to be a Smith member, which implies that the protocol satisfies also two other important criteria, namely *Condorcet criterion* and *Condorcet loser criterion*. Moreover, we show experimentally that the decision chosen through the proposed protocol is a good approximation of the winner. (ii) From the perspective of computation and communication, the proposed protocol enables the agents to make proposals on partial assignments (i.e., assignments of a subset of variables) until all variables are assigned. As a result, it does not require counting candidates as employed in a majority of simple aggregation mechanisms (Lang 2002). It reduces the total number of alternatives that each agent needs to consider in practice, and requires significantly fewer outcome comparisons compared to an exhaustive search in most decision-making instances. (iii) The proposed protocol supports real-world applications. It enables the agents to identify socially preferred decisions without explicitly revealing their preferences, and it can be implemented in a distributed manner. (iv) Last but not least, the proposed protocol is sufficiently general and applicable to most preference representation languages in combinatorial domains. There are no restrictions on the agents' preferences: their preference orders could be complete or partial, and they can extend certain structural properties, e.g., CP-nets, soft constraints, etc.

In our previous work (Li et al. 2011a) we present an efficient protocol for agent-based negotiation in combinatorial domains. Part of the technical framework is similar to the work presented in this paper, in the sense that we also consider the space of possible assignments as a tree, where each level of the tree assigns values to a particular variable. In Li et al. (2011a) we tackle the problem of negotiation and focus on producing Pareto-optimal agreements which is close to the egalitarian solution. However, in this paper we consider the setting of collective decision-making. The aim of this work is to generate efficient social solutions for different Condorcet-consistent rules, which is not a trivial extension of the work in Li et al. (2011a). In particular, while Pareto efficiency is easy to achieve in collective decision-making (e.g., choose the best outcome of an agent as the collective decision), choosing a member of the Smith Set (a generalization of Condorcet consistency) is much more complex in combinatorial domains. Moreover, in Li et al. (2011a) we consider each agent's preference as a complete order that extends flexible preference representation model. In this paper, we further provide solutions for collective decision-making with incomplete preference profiles. Last but not least, in this paper we also introduce a heuristic algorithm to compute an order of variables for decision-making based on the agents preference structures (dependency graphs), which leads towards a more computational efficient process of collective decision-making.

2 Preliminaries

2.1 Combinatorial domains and preferences

Let $\mathbf{V} = \{X_1, \ldots, X_m\}$ be a set of m variables in a combinatorial domain, where each variable X takes values in a *local variable domain* D_X. A variable X is *binary*

if $D_X = \{x, \bar{x}\}$. An alternative is uniquely identified by the combination of variable values. Hence, there are exponentially many possible alternatives (outcomes): $D_{X_1} \times \cdots \times D_{X_m}$, denoted by $\mathcal{X}$. If $\mathbf{X} = \{X_{\sigma_1}, \ldots, X_{\sigma_\ell}\}$ and $\mathbf{X} \subseteq \mathbf{V}$, with $\sigma_1 < \cdots < \sigma_\ell$ then $D_{\mathbf{X}}$ denotes $D_{X_{\sigma_1}} \times \cdots \times D_{X_{\sigma_\ell}}$ and $\mathbf{x}$ denotes an assignment of variable values to $\mathbf{X}$, i.e., $\mathbf{x} \in D_{\mathbf{X}}$. If $\mathbf{X} = \mathbf{V}$, $\mathbf{x}$ is a *complete assignment* (corresponding to an alternative); otherwise $\mathbf{x}$ is called a *partial assignment*. We allow concatenations of disjoint sets of variable values: for instance, let $\mathbf{X} = \{A, B\}$, $\mathbf{Y} = \{C, D\}$, $\mathbf{x} \in D_{\mathbf{X}}$ and $\mathbf{x} = a\bar{b}$, $\mathbf{y} \in D_{\mathbf{Y}}$ and $\mathbf{y} = c\bar{d}$, then $\mathbf{xy}\bar{e}$ denotes an assignment $a\bar{b}c\bar{d}\bar{e}$. If $\mathbf{X} \cup \mathbf{Y} = \mathbf{V}$, we call $\mathbf{xy}$ a completion of assignment $\mathbf{x}$. We denote $\texttt{Comp}(\mathbf{x})$ as the set of completions of $\mathbf{x}$.

A preference order $\mathcal{R}$ on $\mathcal{X}$ is a reflexive, transitive and antisymmetric relation on $\mathcal{X}$ (recall that $\mathcal{R}$ is antisymmetric iff $\forall \mathbf{x}, \mathbf{y} \in \mathcal{X}$, $\mathcal{R}(\mathbf{x}, \mathbf{y})$ and $\mathcal{R}(\mathbf{y}, \mathbf{x})$ implies $\mathbf{x} = \mathbf{y}$). $\mathcal{R}(\mathbf{x}, \mathbf{y})$ is also denoted by $\mathbf{x} \succeq \mathbf{y}$. $\succ$ denotes the strict relation induced from $\mathcal{R}$: $\mathbf{x} \succ \mathbf{y}$ iff $\mathbf{x} \succeq \mathbf{y}$ and *not* $\mathbf{y} \succeq \mathbf{x}$. An order $\mathcal{R}$ is complete (linear order) iff , either $\mathbf{x} \succeq \mathbf{y}$ or $\mathbf{y} \succeq \mathbf{x}$ holds for all $\mathbf{x}, \mathbf{y} \in \mathcal{X}$; otherwise, $\mathcal{R}$ is a partial order, i.e., there exist $\mathbf{x}, \mathbf{y} \in \mathcal{X}$, such that neither $\mathbf{x} \succeq \mathbf{y}$ nor $\mathbf{y} \succeq \mathbf{x}$ holds, written as $\mathbf{x} \bowtie \mathbf{y}$ (incomparable).

In combinatorial domains, direct assessment of the preference relations is usually infeasible due to the exponential outcome space; therefore, several compact languages have been proposed to avoid the exponential blow up. Some of these languages are cardinal, e.g., utility networks (Bacchus et al. 1995) and soft constraints (Domshlak et al. 2006; La et al. 1999) some others are purely qualitative (e.g., CP-nets Boutilier et al. 2004 and CI-nets Bouveret et al. 2009). Some representation models are based on conditional preferences, for instance CP-nets (Boutilier et al. 2004) and its variants; some others are based on propositional logic (or possibly a fragment of it), for instance prioritized goals (Sébastien and Ramón 2004), distance-based goals (Konieczny et al. 2004), weighted goals (Lang 2004), and bidding languages for combinatorial auctions (Boutilier et al. 2001). For brevity, we don't present an exhaustive list here. The reader is referred to Lang (2004) for a detailed survey of compact preference representation languages in combinatorial domains. In the following, we only briefly describe CP-nets, a widely studied graphical language for representing qualitative preferences in combinatorial domains.

A Conditional Preference Network (CP-net) $\mathcal{N}$ (Boutilier et al. 2004) over a set of variables $\mathbf{V}$ is an annotated directed graph $\mathcal{G}_{\mathcal{N}}$, in which nodes stand for the variables. Each node X is annotated with a conditional preference table $CPT(X)$, which associates a strict total order over D_X given every combination values $\mathbf{u}$ of parents $Pa(X)$ in $\mathcal{G}_{\mathcal{N}}$. For instance, let $\mathbf{V} = \{A, B, C\}$, with all three variables being binary, $\mathcal{N}$ be a CP-net whose structural part is the directed acyclic graph $\mathcal{G}_{\mathcal{N}} = \{\mathbf{V}, (C, A), (C, B), (A, B)\}$. This means that the agent's preference over the values of C is unconditional, preference over the values of A (resp. B) is fully determined given the value of C (resp. the values of C and A). The preference statements contained in the conditional preference tables are written with the usual notation, that is, $\bar{a}c : \bar{b} \succ b$ means that when $A = \bar{a}$ and $C = c$, then $B = \bar{b}$ is preferred to $B = b$. Such a relation is denoted by $\bar{b} \succ^{B|\bar{a}c} b$.

A CP-net $\mathcal{N}$ generally induces a partial preorder (written as $\succ_{\mathcal{N}}$), such that $\forall x, x' \in D_X$, $\forall \mathbf{u} \in D_{Pa(X)}$, and $\forall \mathbf{z} \in D_{\mathbf{V}\setminus(Pa(X)\cup\{X\})}$, $x\mathbf{uz} \succ_{\mathcal{N}} x'\mathbf{uz}$ if and only if $x \succ^{X|\mathbf{u}} x'$ (Boutilier et al. 2004). In most literature, the dependency graph $\mathcal{G}_{\mathcal{N}}$ is assumed to be

acyclic. When $\mathcal{G}_{\mathcal{N}}$ is acyclic, $\succ_{\mathcal{N}}$ is transitive and antisymmetric, i.e., a strict partial order.

2.2 Condorcet winner, Smith set, and Condorcet-consistent methods

A preference profile $\mathbf{R} = \langle \mathcal{R}_1, \ldots, \mathcal{R}_n \rangle$ of n agents consists of n preference relations over $\mathcal{X}$; each $\mathcal{R}_i$ represents the preference relation of an agent i. We say a preference profile $\mathbf{R}$ is *complete*, iff $\mathcal{R}_i$ is complete for each $i \leq n$; otherwise, $\mathbf{R}$ is partial. Similarly, we say a preference profile $\mathbf{R}$ is *strict*, iff $\mathcal{R}_i$ is strict for each $i \leq n$. In the typical social choice model, it is assumed that the preference profile $\mathbf{R}$ is strict and complete, i.e., $\mathbf{R}$ consists only strict linear orders. The model then applies an aggregation rule to formulate the choice of the group. When considering different aggregation rules, one of the most popular criteria is Condorcet criterion, which is based on pairwise comparisons and takes into account the majority of agents' preferences. The group of agents chooses an alternative $\mathbf{x}$ over another alternative $\mathbf{y}$ if more than half of the agents prefer $\mathbf{x}$ to $\mathbf{y}$, and the winner, if any, is the candidate that beats each opponent in a pairwise comparison.

Definition 1 Given a preference profile $\mathbf{R}$ over $\mathcal{X}$, let $\#\{i \leq n : \mathbf{x} \succ_i \mathbf{y}\}$ denote the number of agents that prefer an alternative $\mathbf{x}$ over another alternative $\mathbf{y}$, then:

- $\forall \mathbf{x}, \mathbf{y} \in \mathcal{X}$, if $\#\{i \leq n : \mathbf{x} \succ_i \mathbf{y}\} > \frac{n}{2}$, we say $\mathbf{x}$ majority-dominates $\mathbf{y}$, written as $\mathbf{x} \succ_{maj} \mathbf{y}$;
- an alternative $\mathbf{x} \in \mathcal{X}$ is a *Condorcet winner* iff $\forall \mathbf{y} \in \mathcal{X}$ and $\mathbf{y} \neq \mathbf{x}$, $\mathbf{x} \succ_{maj} \mathbf{y}$;
- an alternative $\mathbf{x} \in \mathcal{X}$ is a *Condorcet loser* iff $\forall \mathbf{y} \in \mathcal{X}$ and $\mathbf{y} \neq \mathbf{x}$, $\mathbf{y} \succ_{maj} \mathbf{x}$;

An aggregation rule satisfies the *Condorcet criterion* if it chooses the Condorcet winner when one exists. Any aggregation rule conforming to the Condorcet criterion is known as a Condorcet-consistent rule. Moreover, an aggregation rule satisfies the *Condorcet loser criterion*, if it never chooses a Condorcet loser. In the real-world, collective decision-making often obeys the Condorcet criterion; legislatures, courts, juries, boards of directors and many other institutions decide according to the majority pairwise preferences. In addition to the explicit voting procedures, many groups such as households, cartels and computer networks use aggregation rules that resemble Condorcet-consistent rules (e.g., see Cave and Salant 1995; Fanand and Jin 2009; Epple et al. 2001).

When a Condorcet winner exists, it is unique. However, it is possible for a paradox to occur (known as the Condorcet's Paradox), in which the aggregated preference is cyclic (i.e. it fails to meet transitivity). For example, consider three agents (agent 1, 2 and 3) with the following preferences. Agent 1: $\mathbf{x} \succ \mathbf{y} \succ \mathbf{z}$; agent 2: $\mathbf{y} \succ \mathbf{z} \succ \mathbf{x}$; and agent 3: $\mathbf{z} \succ \mathbf{x} \succ \mathbf{y}$. In such case, a majority of agents (agent 1 and 3) prefer $\mathbf{x}$ to $\mathbf{y}$, another majority of agents (agent 1 and 2) prefer $\mathbf{y}$ to $\mathbf{z}$, and yet another majority of agents (agent 2 and 3) prefer $\mathbf{z}$ to $\mathbf{x}$. Therefore, in this example, the majority rule provides no Condorcet winner.

When a Condorcet winner does not exist, one alternative way to make a collective decision is to look into the Smith set (Ward 1961). The Smith set is the smallest subset of alternatives such that every member of the set is pairwise preferred to every alternative

not in the set, i.e., the smallest nonempty set $\mathcal{S}$ of alternatives such that $\mathbf{x} \succ_{maj} \mathbf{y}$ for all $\mathbf{x} \in \mathcal{S}$ and all $\mathbf{y} \notin \mathcal{S}$. The Smith criterion, sometimes also called generalized Condorcet criterion, states that the aggregation rule should always choose a candidate that is in the Smith set. Obviously, the Smith criterion implies the Condorcet criterion, since if there is a Condorcet winner, then it is the only Smith member. Additionally, the Smith criterion also implies the *Condorcet loser criterion*, as any Condorcet loser is not a member of the Smith set. Finally, we notice that the Smith set can have more than one candidate, either because of pairwise ties, or because of cycles, such as that in a Condorcet's paradox.

In order to make a reasonably good collective decision when a Condorcet winner does not exist, several aggregation rules are proposed in the literature Arrow et al. (2002) and Kenneth (1951). Many of these rules are Condorcet-consistent: they choose the Condorcet winner when one exists. In this paper, we will be mainly concerned with the following Condorcet-consistent aggregation rules: Copeland, Minimax Condorcet (aka. Simpson rule), Schulze, ranked pairs (aka. Tideman method) and Kemeny. Given a complete and strict preference profile $\mathbf{R} = \langle \mathcal{R}_1, \ldots, \mathcal{R}_n \rangle$ and any pair of alternatives $\mathbf{x}, \mathbf{y}$, let $N(\mathbf{x}, \mathbf{y})$ denote the number of agents that rank $\mathbf{x}$ ahead of $\mathbf{y}$ in the profile $\mathbf{R}$, i.e., $N(\mathbf{x}, \mathbf{y}) = \#\{i \leq n : \mathbf{x} \succ_i \mathbf{y}\}$. For the purpose of explanation, we first recall the notion of the *weighted majority graph*. The weighted majority graph (WMG) is a directed graph whose vertices are the alternatives, and there is a directed edge $\mathbf{x} \to \mathbf{y}$ between every pair of vertices where $\mathbf{x} \succ_{maj} \mathbf{y}$ (i.e., $N(\mathbf{x}, \mathbf{y}) > \frac{n}{2}$); the weight on the edge $\mathbf{x} \to \mathbf{y}$ is $N(\mathbf{x}, \mathbf{y})$.

Copeland chooses a candidate that has the maximum number of pairwise victories. Given a preference profile $\mathbf{R}$, the *Copeland score* of an alternative is the number of alternatives it beats in pairwise comparisons. Formally, for an alternative $\mathbf{x}$, let $s(\mathbf{x})$ denote the Copeland score of $\mathbf{x}$, then

$$s(\mathbf{x}) = \#\{\mathbf{y} \in \mathcal{X} : \mathbf{x} \succ_{maj} \mathbf{y}\}$$

A *Copeland winner* is an alternative $\mathbf{x}$ that *maximizes* $s(\mathbf{x})$. In the weighted majority graph of $\mathbf{R}$, the Copeland winners are those candidates with a maximum number of outgoing edges.

Minimax Condorcet also called the *Simpson method*, chooses a candidate that is defeated by the fewest agents in its worst defeat. Given a preference profile $\mathbf{R}$, the *Minimax score* $s(\mathbf{x})$ of an alternative $\mathbf{x}$ is defined by:

$$s(\mathbf{x}) = \max_{\mathbf{y} \in \mathcal{X},\ \mathbf{y} \neq \mathbf{x}} N(\mathbf{y}, \mathbf{x})$$

A *Minimax winner* is an alternative $\mathbf{x}$ that *minimizes* $s(\mathbf{x})$. In a weighted majority graph, the Minimax winners are those candidates that the maximum weight among the incoming edges is minimized.

Schulze chooses a top candidate in a ranking that follows the strongest path between pairs of alternatives. Given a preference profile $\mathbf{R}$, this rule first constructs a weighted majority graph (WMG). In the WMG of a profile $\mathbf{R}$, the weight of a path $Path(\mathbf{x}, \mathbf{y}) = \mathbf{x}_1 (= \mathbf{x}) \to \mathbf{x}_2 \to \cdots \to$

$\mathbf{x}_\ell (= \mathbf{y})$, written as $w(Path(\mathbf{x}, \mathbf{y}))$, is defined as $Min_{i \leq \ell - 1} N(\mathbf{x}_i, \mathbf{x}_{i+1})$. That means, the weight of a path $w(Path(\mathbf{x}, \mathbf{y}))$ is the weight of the weakest edge along the path. A Schulze winner is an alternative $\mathbf{x}$ such that for any other alternative $\mathbf{y}$, $w(Path(\mathbf{x}, \mathbf{y})) > w(Path(\mathbf{y}, \mathbf{x}))$. Such an alternative always exists (Schulze 2011).

Ranked pairs (*aka.* the Tideman method) chooses a top candidate in a ranking constructed following a descending order of supporting votes. This rule first creates an entire ranking of all the candidates, as follows. Given a preference profile $\mathbf{R}$, we consider the pairs of candidates step by step. In each step, we consider a pair of candidates $\mathbf{x}$, $\mathbf{y}$ (with the majority relation $\mathbf{x} \succ_{maj} \mathbf{y}$) that we have not previously considered (as a pair): specifically, we choose the remaining pair with the highest $N(\mathbf{x}, \mathbf{y})$. We then fix the order $\mathbf{x} \succ_{maj} \mathbf{y}$, unless this contradicts previous orders that we fixed (that is, it violates transitivity). We continue until we have considered all pairs of alternatives (hence, in the end, we have a full ranking). Ranked pairs rule then chooses the candidate at the top of the ranking as a winner.

Kemeny chooses a top candidate in the most popular ranking. Given a preference profile $\mathbf{R} = \langle \mathcal{R}_1, \ldots, \mathcal{R}_n \rangle$, for any linear orders $\mathcal{R}_i, \mathcal{R}_j \in \mathbf{R}$ and any pairs of alternatives $\mathbf{x}, \mathbf{y} \in \mathcal{X}$, let $\delta_{\mathbf{x},\mathbf{y}}(\mathcal{R}_i, \mathcal{R}_j)$ equal 1 if $\mathcal{R}_i$ and $\mathcal{R}_j$ agree on which of $\mathbf{x}$ and $\mathbf{y}$ is ranked higher. The Kemeny rule will select all possible linear orders $\mathcal{R}^*$ such that $\sum_{\mathcal{R} \in \mathbf{R}} \sum_{\mathbf{x} \neq \mathbf{y}} \delta_{\mathbf{x},\mathbf{y}}(\mathcal{R}_i, \mathcal{R}_j)$ is maximized (i.e., the rankings that are most popular). The winners for the Kemeny rule are then the top candidates in the most popular rankings.

All the above rules are Condorcet-consistent. Moreover, Copeland, Schulze, Ranked pairs and Kemeny rules also satisfy the Smith criterion, i.e., a winner of the rule is guaranteed to be a member of the Smith set. While Schulze and Ranked pairs are single-winner mechanisms, i.e., the winner of the rule is unique, ties generally occur when using the Copeland, Kemeny, or Minimax Condorcet rule. Finally, notice that the Minimax Condorcet rule fails the Smith criterion. There exist profiles where if we apply the rule, we will choose a winner that is not a Smith member, e.g., a Condorcet loser. However, if the Minimax rule is applied only to the members of the Smith set (i.e., chooses a Minimax winner from the alternatives only in the Smith set), the resulting rule, called Smith/Minimax, satisfies the Smith criterion.

From a computational point of view, computing the above solution concepts from a weighted majority graph is polynomial in the number of alternatives (i.e., exponential in the number of variables in combinatorial domains). Moreover, as shown by Lang (2002), under an assumption of polynomial outcome comparison, checking whether an alternative is a Copeland winner or Minimax winner (not necessarily a Smith member) alone in a combinatorial domain is NP-hard. Furthermore, in many preference representation languages, e.g., CP-nets, outcome comparison is computationally hard (Goldsmith et al. 2008). Therefore, finding good algorithms or protocols to compute or approximate these aggregation rules in combinatorial domains is certainly an important and promising research topic.

2.3 Preference aggregation with incomplete profile

In the previous section, we have discussed several important concepts in preference aggregation with a typical assumption of complete and strict preference profiles. Nonetheless, incompleteness in preferences is not uncommon in many real-life settings (Jérôme et al. 2012). For instance, a user might leave some comparisons between candidates unspecified, because either he doesn't know or he doesn't want to compare some candidates. Brandt, Fischer and Harrenstein (Brandt et al. 2009) study the computational complexity of computing choice sets, without requiring the preference profiles to be complete. They define the Smith set as the minimal dominant subset of alternatives based on the majority-dominant relations induced from the (possibly partial) preference profile. In their definition, every candidate in the Smith set majority-dominates every candidate outside the set, however, the candidates inside the Smith set might not be comparable to each other: there may exist some pairs of alternatives such that the preference profile neither contains enough information to conclude majority dominations nor ties.

An alternate notion of winner in the presence of incomplete profiles is defined in Xia and Conitzer (2011), where the authors define possible winners as the candidates who win for some completion of the partial preference profile, and the necessary winners as the candidates who win in any completion of these partial profiles. A completion of a partial profile $\mathbf{R}$, is a collection of complete (linear) extension for each partial order in that preference profile. A complete extension $\mathcal{R}'$ of a partial order $\mathcal{R} \in \mathbf{R}$, is a *linear* order such that $\mathcal{R} \subseteq \mathcal{R}'$: for any pair of alternatives $\mathbf{x}$, $\mathbf{y}$, if $\mathbf{x} \succ_{\mathcal{R}} \mathbf{y}$, then $\mathbf{x} \succ_{\mathcal{R}'} \mathbf{y}$.

In this paper, we allow the agents to have incomplete preferences. For the sake of simplicity and the purpose of explanation, in the following section, we first introduce our proposed protocol for collective decision-making assuming that the preference profiles are complete and strict. We discuss the detailed techniques to deal with indifference and incompleteness in a later section. Moreover, when the agents' preferences are incomplete, we follow the general definition of Smith set in Brandt et al. (2009).

3 The MDTSC protocol

We now introduce our proposed protocol, called MDTSC, an MDTree-based protocol for Social Choice in combinatorial domains. From a high level, the proposed protocol considers the space of possible assignments as a tree, called the Majority decision tree (MDTree)[2], where each level of the MDTree assigns the values to a particular variable. Following the proposed protocol, the agents iteratively and distributively build the MDTree top-down by proposing on the *leaf*[3] nodes. When we say an agent proposes on (or makes a proposal on) a leaf node, we mean that an agent accepts or votes to continue expanding that node in the MDTree. If a leaf node receives a majority

[2] We call this majority decision tree since the decisions made for the tree expansions are based on the majority rule. That means, a node in the MDTree will be expanded if a majority of agents prefer to.

[3] A leaf node is a node that has zero child nodes.

of agents' proposals, i.e., it is socially preferred to be expanded, it will be expanded to the next level. Once a socially preferred leaf node at the deepest level of the MDTree (i.e., corresponds to a complete assignment) is encounter, it will be added to a set; and the eventual winner of the entire protocol will then be selected from this set, by applying the given social choice rule.

Notice that the agents will only expand those nodes that are socially preferred; the branches of the MDTree in which a majority of agents do not have a more preferred outcome will be pruned out. As such, we do not need to explore the entire outcome space, which is exponential on the number of variables in combinatorial domains.

In the following, we first present the framework for collective decision-making including all the related notions and concepts in Sect. 3.1. Subsequently, in Sect. 3.2 we describe the process of collective decision-making, including the steps and the agents' actions in each iteration when generating the MDTree. We then provide an illustration to demonstrate how the protocol works in Sect. 3.3 and study the theoretical properties of the proposed protocol in Sect. 3.4. For the sake of simplicity, in this section we demonstrate the proposed protocol with complete and strict preference profiles. We provide detailed discussions and solutions on handling incomplete and non-strict preferences in Sect. 5.

3.1 The framework

3.1.1 Majority decision tree (MDTree)

For a collective decision-making problem over m variables $\mathbf{V} = \{X_1, \ldots, X_m\}$, we conceptualize the assignment of the variable values as a tree, called the *majority decision tree* (MDTree). An MDTree assigns values to variables level by level. Assume an order over variables $\sigma = X_{\sigma_1} > \cdots > X_{\sigma_m}$ is given for generating the MDTree $\mathcal{T}$. Then, each depth ℓ ($\ell > 0$) of the tree $\mathcal{T}$ considers the value assignments for variable X_{σ_ℓ}. A node Φ at depth ℓ represents a unique value assignment, denoted by $\mathcal{A}_\Phi$, to the set of variables $\{X_{\sigma_1}, \ldots, X_{\sigma_\ell}\}$ specified by the path from the root to Φ, i.e., $\mathcal{A}_\Phi \in D_{X_{\sigma_1}} \times \cdots \times D_{X_{\sigma_\ell}}$. The depth of $\mathcal{T}$ is m with the root being at depth 0. The root node represents an empty assignment, while a node Φ at depth m represents a unique alternative (outcome), as every variable has been assigned a value, i.e., the cardinality of $|\mathcal{A}_\Phi|$ is m ($|\mathcal{A}_\Phi| = m$). If all the variables in $\mathbf{V}$ are binary, the MDTree $\mathcal{T}$ is a binary tree; otherwise, $\mathcal{T}$ is a k-ary tree, where k is the maximum size of the variable domain: $\forall X \in \mathbf{V},\ |D_X| \leq k$.

3.1.2 Iteratively generating the MDTree

The proposed protocol is iterative. Given a variable order $\sigma = X_{\sigma_1} > \cdots > X_{\sigma_m}$ for collective decision-making, the agents iteratively and distributively generate an MDTree $\mathcal{T}$ top-down by proposing on and expanding the leaf nodes in $\mathcal{T}$.

The proposed protocol starts from the root node of $\mathcal{T}$. We first generate the child nodes of the root, where each branch assigns a unique value to the first variable X_{σ_1}. Then following the protocol, in each iteration, each agent can propose on a leaf node

in $\mathcal{T}$. Once a leaf node Φ at depth ℓ receives a majority of the participating agents' proposals, Φ becomes a socially preferred node[4]:

– If $\ell < m$, Φ is called an *open* node. Formally,

Definition 2 (*Open node*) A node Φ is marked as open if and only if it is a *leaf* node at depth ℓ such that $\ell < m$ and Φ receives a majority of agents' proposals.

An open node is indeed a socially preferred node for expansion. The subtree of Φ will then be expanded and each branch assigns a unique value for the next variable $X_{\sigma_{\ell+1}}$. The child nodes of Φ will be explored by the agents and be available for them to propose on from the next iteration.

In an iteration, more than one open nodes can coexist. For instance, consider three agents 1, 2 and 3 and three nodes Φ_1 Φ_2 and Φ_3. Assume Φ_1 has agent 1's proposal, Φ_2 has agent 2's proposal, and Φ_3 has agent 3's proposal. In an iteration, assume agent 3 proposes on Φ_2, agent 2 proposes on Φ_1, and agent 1 proposes on Φ_3, then Φ_1, Φ_2 and Φ_3 receive a majority of agents' proposals and thus, they are all marked as open in this iteration.

Moreover, once an open node is expanded, it is not an open node any more, as it is no longer a leaf node; and thus, it is not feasible for the agents to make proposals on during later iterations.

– Otherwise, if $\ell = m$, Φ is called a *winning* node. Formally,

Definition 3 (*Winning node*) A node Φ is marked as a winning node if and only if it is a *leaf* node at depth $\ell = m$ (i.e., the deepest level), and Φ receives a majority of agents' proposals.

A winning node, in essence, corresponds to a complete assignment (i.e., an alternative) that receives a majority of agents' proposals. If a node is marked as a winning node, it will not be feasible for the agents to make proposals on during later iterations. Instead, it will be put in a set $\mathcal{W}$, which contains all the winning nodes created so far.

The set of nodes in $\mathcal{W}$ represent the set of possible winning candidates, as the eventual winner of the entire protocol will be selected from $\mathcal{W}$. Notice that the nodes being added into $\mathcal{W}$ will never be taken out of $\mathcal{W}$ in later iterations.

Following the proposed protocol, the agents start to generate the MDTree from the root by proposing on the leaf nodes and expanding the open nodes, until they identify at least one winning nodes. When at least one winning node is encounter, i.e., $\mathcal{W} \neq \emptyset$, one of the winning nodes, denoted by Φ^* will be chosen from $\mathcal{W}$ by applying the given social choice rule to $\mathcal{W}$. Subsequently, the agents will try to converge to this winning node. By "converge", we mean that the agent will then consider Φ^* as a *reference point* (i.e., status quo) for making further proposals in the MDTree. In particular, an agent will continue to propose on another leaf node Φ in a later iteration if and only if it is better than Φ^*, i.e., Φ will potentially bring him to a better outcome than the outcome corresponds to Φ^*. Moreover, when there is no winning node in the

[4] Note that different agents may propose on different leaf nodes (possibly at different depths) in the same iteration. Therefore, when we say a node Φ receives a majority of agents' proposals, these proposals may arrive at Φ in different iterations.

MDTree ($\mathcal{W} = \emptyset$), we set $\Phi^* = NULL$. This can be considered as a disagreement for collective decision-making; and we assume that every agent prefers any other nodes to such a disagreement.

As the agents making further proposals *w.r.t* the chosen reference point Φ^*, more winning nodes might be identified and added into $\mathcal{W}$. Thus, Φ^* might be changed when we add new elements into $\mathcal{W}$. The process of collective decision-making then continues, until all the agents stop making proposals *w.r.t* the current chosen Φ^*. That means, there is no socially more preferred nodes than Φ^*. As such, the outcome corresponds to Φ^* is chosen as the final collective decision (the eventual winner of the proposed protocol).

3.1.3 Best possible alternative and BATCD Strategy

In this paper, we consider a distributed implementation and incomplete information settings. There is no prior information about the preferences of the participating agents. Moreover, the proposals made by an agent during the decision-making process are *invisible* to other agents. In each iteration, the only information that each agent i ($i \leq n$) obtains is a set of leaf nodes that is feasible for her to propose on and a reference point Φ^* (Φ^* is either equal to $NULL$ or it refers to a winning node). In the following sections, we denote $\mathcal{L}_i$ as the set of leaf nodes that are currently feasible for agent i to make proposals on. In an iteration, a node Φ in $\mathcal{T}$ is feasible for an agent i to make a proposal on, i.e., $\Phi \in \mathcal{L}_i$ if and only if:

- it is a *leaf* node and *not* marked as a winning node or open node; and
- agent i has *not* proposed on Φ.

Based on $\mathcal{L}_i$ and Φ^*, agent i will decide on whether or not he will make a proposal, and on which node he will propose. Given Φ^* and when $\Phi^* \neq NULL$, agent i will propose only if there is some node Φ in $\mathcal{L}_i$ that will potentially lead agent i to a more preferred outcome than $\mathcal{A}_{\Phi^*}$. This at least requires that the most preferred alternative from the subtree of Φ is better than $\mathcal{A}_{\Phi^*}$ for agent i. Therefore, in the following, we investigate the optimistic alternative for an individual agent from the branch under a node Φ.

Definition 4 (*Best possible alternative*) Given that agent i's ($i \leq n$) preference is a strict linear order, at each node $\Phi \in \mathcal{L}_i$, agent i has a unique best possible alternative (BPA), denoted by $\texttt{BPA}_i(\Phi)$. It is the optimistic alternative that agent i can obtain from the subtree of Φ, i.e., the best (most preferred) alternative for agent i among the completions of $\mathcal{A}_\Phi$ ($\texttt{Comp}(\mathcal{A}_\Phi)$): $\forall \mathbf{y} \in \texttt{Comp}(\mathcal{A}_\Phi)$ and $\mathbf{y} \neq \texttt{BPA}_i(\Phi)$, $\texttt{BPA}_i(\Phi) \succ_i \mathbf{y}$.

For those nodes Φ that are in the deepest level m, the best possible alternative $\texttt{BPA}_i(\Phi)$ of any agent i is equal to the assignment $\mathcal{A}_\Phi$, i.e., $\forall i \leq n, \texttt{BPA}_i(\Phi) = \mathcal{A}_\Phi$ when $|\mathcal{A}_\Phi| = m$. For instance, assume $\mathbf{V} = \{A, B, C\}$, all variables are binary: $D_A = \{a, \bar{a}\}$, $D_B = \{b, \bar{b}\}$ and $D_C = \{c, \bar{c}\}$. Consider a node Φ with assignment a (resp. $\bar{a}\bar{b}$), the completions of $\mathcal{A}_\Phi$ are $\{abc, ab\bar{c}, a\bar{b}c, a\bar{b}\bar{c}\}$ (resp. $\{\bar{a}\bar{b}c, \bar{a}\bar{b}\bar{c}\}$). If an agent i's preference is represented by a CP-net: $c : \bar{a} \succ a$, $\bar{c} : a \succ \bar{a}$; $b \succ \bar{b}$; and $c \succ \bar{c}$. Then the BPA of Φ is abc (resp. $\bar{a}\bar{b}c$), because as induced from that CP-net,

$abc \succ_i ab\bar{c}$, $abc \succ_i a\bar{b}c$ and $abc \succ_i a\bar{b}\bar{c}$ (resp. $\bar{a}\bar{b}c \succ_i \bar{a}\bar{b}\bar{c}$). Moreover, the BPA of the root node of agent i corresponds to the best (most preferred) alternative of agent i in the entire alternative space. With the aforementioned CP-net, it is $\bar{a}bc$. Notice that when agent i's preference is complete and strict, or if it is represented by an acyclic CP-net, the BPA of a node is always unique. Otherwise, there might be more than one BPAs of a node which are incomparable or indifferent to each other for agent i. We'll further discuss incomplete and nonstrict preference handling in Sect. 5.

Consequently, given the reference point Φ^* (when $\Phi^* \neq NULL$), if $\exists \texttt{BPA}_i(\Phi) \in \mathcal{L}_i$, such that $\texttt{BPA}_i(\Phi) \succ_i \mathcal{A}_{\Phi^*}$, agent i will propose in the current iteration. Moreover, during the decision-making process, an agent can always go backtrack to a node in a different branch if the BPA of the current node is less preferred than that of the other node. Therefore, we can assume that the agents use an optimistic strategy for making proposals in the MDTree. That means, if agent i decides to propose, he will choose a node Φ that $\texttt{BPA}_i(\Phi)$ is the most preferred (non-dominated) among those of the nodes in $\mathcal{L}_i$. We first define the preference relation of an individual agent on the leaf nodes based on the agent's optimistic evaluation (i.e., the BPA) of each leaf node.

Definition 5 (*Preference relations between leaf nodes*) Given a pair of leaf nodes Φ and Φ', we say an agent i prefers Φ over Φ' (written as $\Phi \succ_i \Phi'$) iff agent i prefers the BPA of Φ over the BPA of Φ'. Formally, $\forall i \leq n$ and $\forall \Phi,\ \Phi' \in \mathcal{L}_i$, $\Phi \succ_i \Phi'$ iff $\texttt{BPA}_i(\Phi) \succ_i \texttt{BPA}_i(\Phi')$.

We consider the following BATCD strategy (Best alternative to a collective decision):

Definition 6 (*BATCD strategy*) Given a list of feasible leaf nodes $\mathcal{L}_i$ and Φ^* in the current iteration, agent i ($i \leq n$) will make a proposal on a node $\Phi \in \mathcal{L}_i$ if and only if:

- $\forall \Phi' \in \mathcal{L}_i$ and $\Phi' \neq \Phi$, $\Phi \succ_i \Phi'$; and
- $\Phi^* = NULL$ or $\Phi \succ_i \Phi^*$.

As an implementation, each agent i can maintain a priority queue consisting of nodes, known as the *fringe*. A fringe is a *sorted* list of feasible nodes $\mathcal{L}_i$ *w.r.t.* agent i's preferences. That is, the more preferred a node Φ is, the higher priority it is in agent i's fringe. In each iteration, given Φ^* (possibly $NULL$), agent i considers the highest priority node (i.e., the most prefer node in fringe). Once a node in $\mathcal{T}$ is expanded, each agent will add the new child nodes into his fringe following his preference over those nodes. Moreover, infeasible nodes will be removed from his fringe. As we mentioned before, these infeasible nodes include the node that he proposes on (as an agent can propose on a node at most once), and the created open nodes and winning nodes.

Remark 1 We emphasize here that the agents' preferences can be represented by most compact languages in combinatorial domains, and different agents can use different representation languages. Moreover, the agents only need to conduct a few outcome

optimization queries[5] (for computing the BPAs) and dominance queries[6] when adding new nodes into their fringes, while they do not need to reason about the entire preference ordering during the decision-making process.[7]

Nonetheless, the computational complexity of outcome optimization queries and dominance queries varies with different preference representation language used. For instance, in the context of acyclic CP-nets, computing the BPA of a node for an agent is similar to the individual outcome optimization with constraints (i.e., the values assigned to the variables along the path from the root node to the current node being fixed). We simply need to sweep through the network from ancestors to descendants, assigning the most preferred value to each remaining variable X (i.e., the variable that has not been assigned a value along the path) respecting to the parent context. Therefore, it is linear in the size of the preference network. However, the computation cost of individual dominance testing in general CP-nets is hard. With the exception of special cases such as CP-nets with tree or polytree structured conditional dependencies, testing dominance relations in acyclic CP-nets is still NP-hard (Boutilier et al. 2004).[8] Therefore, it is important to reduce the number of dominance queries in collective decision-making.

Remark 2 At each node of the MDTree, we consider the best possible alternative rather than using a pessimistic strategy by considering the worst possible alternative. This is because using a pessimistic strategy may results in inefficient outcomes. Take a simple example, consider a two-agent preference profile $\mathbf{R} = \langle \mathcal{R}_1, \mathcal{R}_2 \rangle$ over the combination values of two binary variables A and B, where $\mathcal{R}_1 = \mathcal{R}_2 = ab \succ \bar{a}b \succ \bar{a}\bar{b} \succ a\bar{b}$. Assume we generate the MDTree following the variable order $\sigma = A > B$. Using a pessimistic strategy, both agents' worst possible alternative of branch of a is $a\bar{b}$, and of branch $\bar{a}$ is $\bar{a}\bar{b}$. As both agents prefer $\bar{a}\bar{b}$ to $a\bar{b}$, they will propose on the node with the partial assignment $\bar{a}$, which makes the node with $\bar{a}$ being expanded. Finally, both agent will propose on the node with assignment $\bar{a}\bar{b}$, as both of them prefers $\bar{a}\bar{b}$ to $\bar{a}b$. Hence, the resulting final agreement is $\bar{a}\bar{b}$. However, as for both agents $ab \succ \bar{a}\bar{b}$, the final chosen alternative $\bar{a}\bar{b}$ is an inefficient outcome.

3.2 The process of collective decision-making

We now present the process of collective decision-making using the proposed `MDTSC` protocol. From a broad view, the proposed `MDTSC` protocol is an iterative procedure, where the agents iteratively make proposals on the nodes in $\mathcal{T}$ and as a result, the nodes in $\mathcal{T}$ are expanded and explored. Once there exists one or some winning nodes, the

[5] An outcome optimization query determines the set of non-dominated outcomes among the feasible outcome space with respect to an agent's preferences.

[6] A dominance query, given two alternatives o and o', asks whether o is preferred to o' with respect to an agent's preferences.

[7] Reasoning about the induced preference ordering over the outcome space may require exponential number of dominance queries.

[8] When the CP tables are represented in a compact way, dominance testing in binary-valued acyclic CP-nets are PSPACE-complete (Goldsmith et al. 2008).

Protocol 1: MDTSC

Input: $\sigma = X_{\sigma_1} > \cdots > X_{\sigma_m}$, an order over variables; r, a social choice rule
Output: $\mathcal{A}_{\Phi^*}$, a collective decision (outcome);

1 Initiate an MDTree $\mathcal{T}$ with a root node;
// Expand the root node by creating the child nodes with each branch assigns a unique value to the first variable X_{σ_1}.
2 **foreach** $x_{\sigma_1} \in D_{X_{\sigma_1}}$ **do**
3 $\quad \Phi \leftarrow$ CreateNode($root, x_{\sigma_1}$); AppendTo($\mathcal{T}, \Phi$);
4 **end**
5 $\Phi^* \leftarrow NULL$; $\mathcal{W} \leftarrow \emptyset$;
6 **repeat**
// Step 1: Offering
7 Each agent i makes a a proposal if there exists a node in $\mathcal{L}_i$ that is preferred to Φ^*;
// Step 2: Expanding
8 **foreach** *open node* Φ **do**
9 $\quad k \leftarrow |\mathcal{A}_\Phi| + 1$;
// Expand Φ and create the child nodes of Φ with each branch assigns a unique value to the next variable X_{σ_k}.
10 $\quad$ **foreach** $x_{\sigma_k} \in D_{X_{\sigma_k}}$ **do**
11 $\quad\quad \Phi' \leftarrow$ CreateNode(Φ, x_{σ_k}); AppendTo($\mathcal{T}, \Phi'$);
12 $\quad$ **end**
13 **end**
// Step 3: Choosing Φ^*
14 **add** all winning nodes created **to** $\mathcal{W}$;
// chooses a winner in $\mathcal{W}$ according to r.
15 $\Phi^* \leftarrow$ SC($\mathcal{W}, r$);
16 **until** *all agents do not propose in Step 1*;
17 **return** $\mathcal{A}_{\Phi^*}$;

agents then try to converge to a chosen winner in the set of winning nodes, i.e., Φ^*, until they could not find any other alternative that is socially more preferred. Then they stop making proposals, and the process ends with a collective decision $\mathcal{A}_{\Phi^*}$. Notice that Φ^* is originally $NULL$ (representing a disagreement), and is dynamically changing during the decision-making process, as the agents may identify other different socially more preferred winning nodes.

Protocol 1 provides an algorithmic implementation of the collective decision-making process. The input of the protocol is an order $\sigma = X_{\sigma_1} > \cdots > X_{\sigma_m}$ over the domain variables following which the MDTree will be formed. Without any knowledge about the agents' preferences, this order can be randomly chosen or decided by a neutral third party. Notice that if a different order is chosen to generate the MDTree, it may lead to a different final decision, but as we will prove later, the final decision chosen by the MDTSC protocol is guaranteed to be a Smith member. Nonetheless, it may also affect the size of the MDTree, and thus, the total number of alternatives that each agent needs to consider during the decision-making process (i.e., the total number of leaf nodes being explored in the MDTree) .

Initially, an MDTree $\mathcal{T}$ is initiated with a root node. We create and add all possible $|D_{X_{\sigma_1}}|$ child nodes (each branch assigns a distinct value to the variable X_{σ_1}) to $\mathcal{T}$ (lines 2–4). Each agent i ($i \leq n$) adds these nodes into his fringe *w.r.t* his preferences.

Φ^* is assigned $NULL$ and the set of winning nodes $\mathcal{W}$ is assigned $\emptyset$ (line 5). Each iteration of the MDTSC protocol then performs the following three steps:

Step 1: Offering. Each agent i ($i \leq n$) proposes on a feasible *leaf* node $\Phi \in \mathcal{L}_i$ (i.e., the most preferred node in $\mathcal{L}_i$) and removes Φ from $\mathcal{L}_i$, unless:

- agent i has made a proposal on every feasible leaf node, i.e., his fringe currently is empty ($\mathcal{L}_i = \emptyset$); or
- none of the nodes in $\mathcal{L}_i$ has a BPA that is preferred to $\mathcal{A}_{\Phi^*}$: $\forall \Phi \in \mathcal{L}_i, \mathcal{A}_{\Phi^*} \succ_i \mathtt{BPA}_i(\Phi)$.

After this step, some leaf nodes in $\mathcal{T}$ will become open nodes (if its depth is smaller than m and it receives a majority of agents' proposals); and some leaf nodes will become winning nodes (if it is at depth m and receives a majority of agents' proposals). Each agent then removes all created open nodes and winning nodes from his fringe $\mathcal{L}_i$, as they are no longer feasible for making proposals on.

Step 2: Expanding. We expand all the open nodes created in the current iteration (lines 8–13). For an open node Φ at depth ℓ ($\ell < m$), let $k = \ell + 1$, it will be expanded with every possible value x_{σ_k} of the next variable X_{σ_k}. Each agent i then adds these child nodes to $\mathcal{L}_i$ for consideration.

Step 3: Choosing Φ^*. All the winning nodes created will be added into a set $\mathcal{W}$ (line 14). Then in the third step, we apply a given social rule r to the set of alternatives $\mathcal{W}$ and choose a winner (computed by the social choice function $\mathtt{SC}(\mathcal{W}, r)$) to be the reference point Φ^* for convergence (line 15).

The process then goes back to Step 1 and continues, until all the agents stop making proposals in Step 1 (line 16). If so, the current chosen Φ^* is the eventual winner and the protocol returns the corresponding outcome as the final collective decision (line 17).

Notice that during the entire decision-making process, only the alternatives in $\mathcal{W}$ require direct pairwise comparisons.

3.3 Illustration

Now, we demonstrate the execution of MDTSC protocol with an example. Assume agents 1, 2 and 3 are making decisions over a set of three binary-valued variables $\mathbf{V} = \{A, B, C\}$. The domains are $D_A = \{a, \bar{a}\}$, $D_B = \{b, \bar{b}\}$ and $D_C = \{c, \bar{c}\}$, respectively. For the sake of simplicity, we assume that the agents have strict linear orders over $\mathcal{X}$, which are depicted in Fig. 1. We consider the Smith/Minimax aggregation rule, i.e., $r = Smith/Minimax$. In essence, these linear orders extend the following CP-nets respectively:

- agent 1: $c : \bar{a} \succ a, \bar{c} : a \succ \bar{a}; b \succ \bar{b}; c \succ \bar{c}$;
- agent 2: $a \succ \bar{a}; \bar{b} \succ b; b : \bar{c} \succ c, \bar{b} : c \succ \bar{c}$;
- agent 3: $b : \bar{a} \succ a, \bar{b} : a \succ \bar{a}; b \succ \bar{b}; a : c \succ \bar{c}, \bar{a} : \bar{c} \succ c$.

As all variables are binary, the MDTree is a binary tree. Figure 2 shows the formation of the MDTree $\mathcal{T}$. Each node Φ in $\mathcal{T}$ associates with a proposal table, in which the

$\bar{a}bc \succ abc \succ ab\bar{c} \succ \bar{a}b\bar{c} \succ \bar{a}\bar{b}c \succ a\bar{b}c \succ a\bar{b}\bar{c} \succ \bar{a}\bar{b}\bar{c}$
(a) Agent 1

$a\bar{b}c \succ \bar{a}\bar{b}c \succ a\bar{b}\bar{c} \succ \bar{a}\bar{b}\bar{c} \succ ab\bar{c} \succ \bar{a}b\bar{c} \succ abc \succ \bar{a}bc$
(b) Agent 2

$\bar{a}b\bar{c} \succ \bar{a}bc \succ abc \succ ab\bar{c} \succ a\bar{b}c \succ a\bar{b}\bar{c} \succ \bar{a}\bar{b}\bar{c} \succ \bar{a}\bar{b}c$
(c) Agent 3

Fig. 1 Agents' preferences

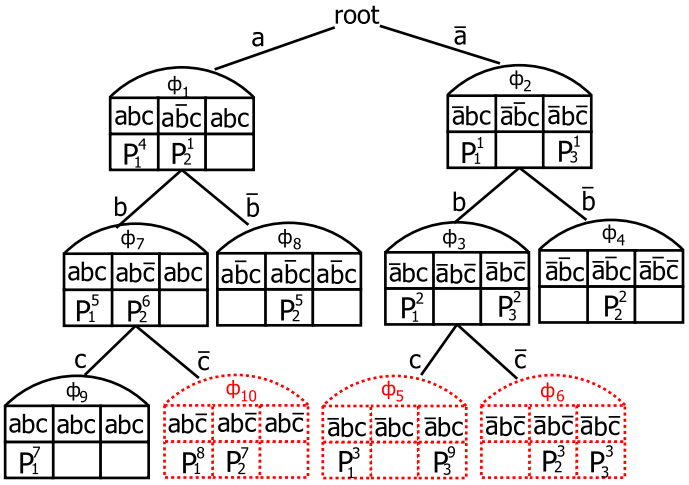

Fig. 2 MDTree $\mathcal{T}$

second row of the table displays the proposals that the agents make on that node: the 1st, 2nd and 3rd column depicts the proposal that agent 1, agent 2 and agent 3 makes, respectively; and the proposal made by agent j in the ith iteration is denoted by P_j^i. For explanation purposes, we also attach the best possible alternative (BPA) of each agent on each node in the first row of the table: the 1st, 2nd and 3rd column depicts the BPA of agent 1, agent 2 and agent 3, respectively. However, notice that the information including the proposal that an agent makes and the BPA of a node of an agent is his private information and invisible to other agents. The nodes in $\mathcal{T}$ in red dashed boxes are the winning nodes in $\mathcal{W}$. Figure 3 provides an illustration of the ongoing changes occurring in each agent's fringe. The number I_j denotes the jth iteration, and the bullets in different colors denote whether that agent makes a proposal in that iteration: "$\circ$" denotes that the agent proposes, and "$\bullet$" means the agent does *not* make any proposal in that iteration.

Suppose that an order $\sigma = A > B > C$ is chosen for the formation of an MDTree $\mathcal{T}$. Initially, a root node as well as two child nodes Φ_1, Φ_2 with assignment a and $\bar{a}$, respectively are created in $\mathcal{T}$. For agent 1, $\mathtt{BPA}_1(\Phi_1) = abc$, $\mathtt{BPA}_1(\Phi_2) = \bar{a}bc$. Since $\bar{a}bc \succ_1 abc$ (see Fig. 1a), the order of nodes in agent 1's fringe is Φ_2–Φ_1 (i.e., agent 1 prefers Φ_2–Φ_1). Similarly, agent 2's fringe is Φ_1–Φ_2, and agent 3's fringe is Φ_2–Φ_1.

	Φ^*	Agent 1	Agent 2	Agent 3
I_1 :		○ Φ_2–Φ_1	○ Φ_1–Φ_2	○ Φ_2–Φ_1
I_2 :		○ Φ_3–Φ_1–Φ_4	○ Φ_4–Φ_3	○ Φ_3–Φ_1–Φ_4
I_3 :		○ Φ_5–Φ_1–Φ_6–Φ_4	○ Φ_6–Φ_5	○ Φ_6–Φ_5–Φ_1–Φ_4
I_4 :	Φ_6	○ Φ_1–Φ_4	● Φ_5	● Φ_5–Φ_1– Φ_4
I_5 :	Φ_6	○ Φ_7–Φ_4–Φ_8	○ Φ_8–Φ_7–Φ_5	● Φ_5–Φ_7–Φ_8–Φ_4
I_6 :	Φ_6	● Φ_4–Φ_8	○ Φ_7–Φ_5	● Φ_5–Φ_7– ϕ –Φ_4
I_7 :	Φ_6	○ Φ_9–Φ_{10}–Φ_4–Φ_8	○ Φ_{10}–Φ_9–Φ_5	● Φ_5–Φ_9–Φ_{10}–Φ_8–Φ_4
I_8 :	Φ_6	○ Φ_{10}–Φ_4–Φ_8	● Φ_9–Φ_5	● Φ_5–Φ_9–Φ_{10}–Φ_8–Φ_4
I_9 :	Φ_{10}	● Φ_4–Φ_8	● Φ_9–Φ_5	○ Φ_5–Φ_9–Φ_8–Φ_4
I_{10}:	Φ_6	● Φ_4– Φ_8	● Φ_9–Φ_5	● Φ_9–Φ_8–Φ_4

Fig. 3 Agents' fringes

In the **1***st* iteration, agent 1 and 3 propose on Φ_2 (the first node in their fringes), agent 2 proposes on Φ_1 in Step 1. Thus, Φ_2 is marked as open as it received a majority of agents' proposals. Φ_2 is then expanded with the possible values $\{b, \bar{b}\}$ of the second variable B. Two child nodes Φ_3, Φ_4 are created in $\mathcal{T}$, and they are added into each agent's fringe (see Fig. 3). Notice that agent 2's fringe does not contain Φ_2, although it has not made a proposal on Φ_2. This is because Φ_2 is marked as open and is expanded in this iteration. Therefore, it is not feasible for agent 2 to make a proposal on.

Similarly in the **2***nd* iteration, agent 1 and 3 propose on Φ_3, agent 2 proposes on Φ_4. Φ_3 becomes open and two child nodes Φ_5 and Φ_6 are created.

In the **3***rd* iteration, agent 1 proposes on Φ_5, agent 2 and 3 propose on Φ_6. Φ_6 becomes a winning node and is added into $\mathcal{W}$. Since it is the only member in $\mathcal{W}$, the node chosen for convergence is $\Phi^* = \Phi_6$. Notice that agent 1 will remove the winning node Φ_6 from his fringe, as it is not a feasible node any more.

In the **4***th* iteration, both agent 2 and 3 do not propose, as $\mathcal{A}_{\Phi^*}$ is preferred to the BPA of the most preferred node in their fringe: $\mathcal{A}_{\Phi^*} \succ_2 \texttt{BPA}_2(\Phi_5)$ and $\mathcal{A}_{\Phi^*} \succ_3 \texttt{BPA}_3(\Phi_5)$. Agent 1 proposes on Φ_1 because $\texttt{BPA}_1(\Phi_1) \succ_1 \mathcal{A}_{\Phi^*}$. Φ_1 is then marked as open and will be expanded in Step 2. Two child nodes Φ_7 and Φ_8 are created in $\mathcal{T}$. Agent 1, 2, 3 then add these nodes into their fringes respectively.

In the **5***th* iteration, agent 1 proposes on node Φ_7, agent 2 proposes on Φ_8, and agent 3 still does not make any proposal.

In the **6***th* iteration, agent 1 and agent 3 do not make proposals, agent 2 proposes on Φ_7. Φ_7 becomes open and is expanded. Two child nodes Φ_9 and Φ_{10} are created in $\mathcal{T}$ and added into each agent's fringe.

In the **7***th* iteration, agent 1 proposes on Φ_9, agent 2 proposes on Φ_{10}, and agent 3 does not make any proposal.

In the **8***th* iteration, agent 2 and agent 3 do not make any proposal, agent 1 proposes on Φ_{10}. As such, Φ_{10} becomes another winning node. Φ_{10} is added into $\mathcal{W}$. Φ_{10} is chosen to be the node for convergence $\Phi^* = \Phi_{10}$ (since it is the Smith/Minimax solution in $\mathcal{W} = \{\Phi_6, \Phi_{10}\}$: $\mathcal{A}_{\Phi_{10}} \succ_{maj} \mathcal{A}_{\Phi_6}$: agent 1 and 2 prefer $\mathcal{A}_{\Phi_{10}}$ over $\mathcal{A}_{\Phi_6}$).

In the **9***th* iteration, agent 1 and agent 2 do not make proposal, agent 3 proposes on Φ_5, since he prefers Φ_5 over Φ_{10}. Φ_5 becomes a winning node. Therefore, currently

$\mathcal{W} = \{\Phi_6, \Phi_{10}, \Phi_5\}$. In this example, Φ_5, Φ_6 and Φ_{10} form a circle: $\mathcal{A}_{\Phi_6} \succ_{maj} \mathcal{A}_{\Phi_5}$, $\mathcal{A}_{\Phi_5} \succ_{maj} \mathcal{A}_{\Phi_{10}}$, and $\mathcal{A}_{\Phi_{10}} \succ_{maj} \mathcal{A}_{\Phi_6}$, and the maximum defeats of all of them are two. Consequently, they are all Smith/Minimax candidates in $\mathcal{W}$. We randomly choose one of them for convergence, for instance, consider $\Phi^* = \Phi_6$.

In the **10***th* iteration, all three agents stop making proposals: $\forall i \leq n, \forall \Phi \in \mathcal{L}_i$, $\texttt{BPA}_i(\Phi) \not\succ \mathcal{A}_{\Phi_6}$. The process ends and returns $\mathcal{A}_{\Phi_6} = \bar{a}b\bar{c}$ as the final collective decision. In this illustrative example, $\bar{a}b\bar{c}$ is an actual Smith/Minimax candidate in the alternative space. Using the MDTSC protocol, the number of alternatives that each agent needs to consider (i.e., the number of leaf nodes in the MDTree) is 6. Moreover, the process only requires pairwise comparisons on a set of three alternatives $\mathcal{W} = \{\mathcal{A}_{\Phi_5}, \mathcal{A}_{\Phi_6}, \mathcal{A}_{\Phi_{10}}\}$, which is less than half of the alternative space.

3.4 Properties of MDTSC protocol

To prove that the final chosen decision using the MDTSC protocol is a member of the Smith set, we first show the following proposition and corollary. In Proposition 1, we prove that for individual preferences, if an agent prefers a winning node Φ^* to another node Φ, then she prefers the outcome corresponds to Φ^* to any outcome following the branch of Φ. Extending to majority preferences, Corollary 1 then states that if a winning node Φ^* is majority preferred to another node Φ, then the outcome corresponds to the winning node is also majority preferred to any outcome following the branch of Φ.

Proposition 1 *For an individual agent i ($i \leq n$), given a* winning *node Φ^* and a node Φ (not necessarily at depth m), if $\Phi^* \succ_i \Phi$, then:* $\forall \mathbf{x} \in \texttt{Comp}(\mathcal{A}_\Phi)$, $\mathcal{A}_{\Phi^*} \succ_i \mathbf{x}$.

Proof According to Definition 5, for agent i, if $\Phi^* \succ_i \Phi$, then $\mathcal{A}_{\Phi^*} (= \texttt{BPA}_i(\Phi^*)) \succ_i \texttt{BPA}_i(\Phi)$. Since, $\forall \mathbf{x} \in \texttt{Comp}(\mathcal{A}_\Phi)$ and $\mathbf{x} \neq \texttt{BPA}_i(\Phi)$, $\texttt{BPA}_i(\Phi) \succ_i \mathbf{x}$ and individual preference is transitive, assume by contradiction that $\mathbf{x} \succ_i \mathcal{A}_{\Phi^*}$, then we will have $\texttt{BPA}_i(\Phi) \succ_i \mathbf{x} \succ_i \mathcal{A}_{\Phi^*}$, contradicting the fact that $\mathcal{A}_{\Phi^*} \succ_i \texttt{BPA}_i(\Phi)$. □

Let $\#\{i \leq n : \Phi \succ_i \Phi'\}$ denote the number of agents that prefer Φ to Φ', in the following section we say a node Φ majority dominates another node Φ', written as $\Phi \succ_{maj} \Phi'$, if $\#\{i \leq n : \Phi \succ_i \Phi'\} > \frac{n}{2}$.

Corollary 1 *Given a* winning *node Φ^* and a node Φ (not necessarily at depth m), if $\Phi^* \succ_{maj} \Phi$, then:* $\forall \mathbf{x} \in \texttt{Comp}(\mathcal{A}_\Phi)$, $\mathcal{A}_{\Phi^*} \succ_{maj} \mathbf{x}$.

Let $\mathcal{L}$ denote the union of $\mathcal{L}_i$, $i \leq n$ ($\mathcal{L} = \bigcup_{i \leq n} \mathcal{L}_i$); $\mathcal{C}_\mathcal{W}$ denote the outcomes specified by the nodes in $\mathcal{W}$, i.e., $\mathcal{C}_\mathcal{W} = \{\mathcal{A}_\Phi | \Phi \in \mathcal{W}\}$; and $\mathcal{C}_\mathcal{L}$ denote all possible completions to the partial assignments specified by the nodes in $\mathcal{L}$, i.e., $\mathcal{C}_\mathcal{L} = \bigcup_{\Phi \in \mathcal{L}} \texttt{Comp}(\mathcal{A}_\Phi)$. Now, we prove that the proposed protocol satisfies the Smith criterion.

Theorem 1 *The final collective decision $\mathcal{A}_{\Phi^*}$ chosen by MDTSC protocol with $r \in \{Copeland, Smith/Minimax, Schulze, Rankedpairs, Kemeny\}$ is guaranteed to be a member of the Smith set.*

Proof $\mathcal{A}_{\Phi^*}$ is a Smith member in $\mathcal{C}_{\mathcal{W}}$ (since $\forall r \in \{Copeland, Smith/Minimax, Schulze, Rankedpairs, Kemeny\}, r$ satisfies the Smith criterion). Let $\mathcal{S}$ be the Smith set in $\mathcal{W}$, and $\mathcal{C}_{\mathcal{S}}$ be the outcomes corresponding to the nodes in $\mathcal{S}$. Then, $\forall \mathbf{x} \in \mathcal{C}_{\mathcal{S}}$ and $\mathbf{x} \neq \mathcal{A}_{\Phi^*}$, either there is a majority preference cycle or there is a majority preference tie between $\mathcal{A}_{\Phi^*}$ and $\mathbf{x}$. In either case, if $\mathbf{x} \in \mathcal{C}_{\mathcal{S}} \setminus \{\mathcal{A}_{\Phi^*}\}$ happens to be a Smith member in $\mathcal{X}$, then $\mathcal{A}_{\Phi^*}$ must also be a Smith member in $\mathcal{X}$.

As $\mathcal{C}_{\mathcal{S}}$ is the Smith set in $\mathcal{C}_{\mathcal{W}}$, $\forall \mathbf{x} \in \mathcal{C}_{\mathcal{W}} \setminus \mathcal{C}_{\mathcal{S}}, \mathcal{A}_{\Phi^*} \succ_{maj} \mathbf{x}$. Therefore, $\forall \mathbf{x} \in \mathcal{C}_{\mathcal{W}} \setminus \mathcal{C}_{\mathcal{S}}$, if $\mathbf{x}$ happens to be a Smith member in $\mathcal{X}$, then $\mathcal{A}_{\Phi^*}$ must also be a Smith member in $\mathcal{X}$.

Moreover, when all the agents stop making proposals and stick with Φ^*, for any node $\Phi \in \mathcal{L}$, $\Phi^* \succ_{maj} \Phi$ as a majority of agents chooses to stick with Φ^* instead of exploring Φ. According to Corollary 1, $\forall \mathbf{x} \in \mathtt{Comp}(\mathcal{A}_{\Phi}), \mathcal{A}_{\Phi^*} \succ_{maj} \mathbf{x}$. Hence, $\forall \mathbf{x} \in \mathcal{C}_{\mathcal{L}}, \mathcal{A}_{\Phi^*} \succ_{maj} \mathbf{x}$. Therefore, $\forall \mathbf{x} \in \mathcal{C}_{\mathcal{L}}$, if $\mathbf{x}$ happens to be a Smith member in $\mathcal{X}$, then $\mathcal{A}_{\Phi^*}$ must also be a Smith member in $\mathcal{X}$.

Finally, from the above analysis we can conclude that $\forall \mathbf{x} \in \mathcal{X} \setminus \{\mathcal{A}_{\Phi^*}\}$, if $\mathbf{x}$ happens to be a Smith member in $\mathcal{X}$, then $\mathcal{A}_{\Phi^*}$ must also be a Smith member in $\mathcal{X}$. Hence $\mathcal{A}_{\Phi^*}$ must be a Smith member in $\mathcal{X}$. □

Theorem 2 *Let $|\mathcal{C}_{\mathcal{W}}|$ denote the number of candidates in $\mathcal{W}$, and $s^{\mathcal{C}_{\mathcal{W}}}_{Copeland}(\mathcal{A}_{\Phi^*})$ denote the Copeland score of $\mathcal{A}_{\Phi^*}$ in $\mathcal{C}_{\mathcal{W}}$, i.e., the number of candidates in $\mathcal{C}_{\mathcal{W}}$ that are majority dominated by $\mathcal{A}_{\Phi^*}$. If $r = Copeland$, the distance between the Copeland score of the chosen decision $\mathcal{A}_{\Phi^*}$ and the Copeland winner is bounded by $|\mathcal{C}_{\mathcal{W}}| - s^{\mathcal{C}_{\mathcal{W}}}_{Copeland}(\mathcal{A}_{\Phi^*}) - 2$.*

Proof As $\mathcal{A}_{\Phi^*}$ is a Copeland winner in $\mathcal{C}_{\mathcal{W}}$ and it dominates every outcome in $\mathcal{C}_{\mathcal{L}}$, the Copeland score of $\mathcal{A}_{\Phi^*}$ in $\mathcal{X} = \mathcal{C}_{\mathcal{W}} \cup \mathcal{C}_{\mathcal{L}}$ is then $s^{\mathcal{C}_{\mathcal{W}}}_{Copeland}(\mathcal{A}_{\Phi^*}) + |\mathcal{C}_{\mathcal{L}}|$. If $\mathcal{A}_{\Phi^*}$ is not a Copeland winner, then any other outcome $\mathbf{x} \in \mathcal{C}_{\mathcal{W}}$ must not be a Copeland winner. This is because $s^{\mathcal{C}_{\mathcal{W}}}_{Copeland}(\mathbf{x}) \leq s^{\mathcal{C}_{\mathcal{W}}}_{Copeland}(\mathcal{A}_{\Phi^*})$ and $\mathbf{x}$ might not dominate all the alternatives in $\mathcal{C}_{\mathcal{L}}$.

If $\mathbf{x} \notin \mathcal{C}_{\mathcal{W}}$ is a Copeland winner, the maximum number of alternatives it dominates in $\mathcal{C}_{\mathcal{W}}$ is $|\mathcal{C}_{\mathcal{W}}| - 1$, and the maximum number of alternatives it dominates in $\mathcal{C}_{\mathcal{L}}$ is $|\mathcal{C}_{\mathcal{L}}| - 1$. Therefore, the upper bound of the difference between the Copeland scores of the Copeland winner and the chosen candidate $\mathcal{A}_{\Phi^*}$ is $(|\mathcal{C}_{\mathcal{W}}| - 1 + |\mathcal{C}_{\mathcal{L}}| - 1) - (s^{\mathcal{C}_{\mathcal{W}}}_{Copeland}(\mathcal{A}_{\Phi^*}) + |\mathcal{C}_{\mathcal{L}}|) = |\mathcal{C}_{\mathcal{W}}| - s^{\mathcal{C}_{\mathcal{W}}}_{Copeland}(\mathcal{A}_{\Phi^*}) - 2$. □

Theorem 3 *Let $s^{\mathcal{C}_{\mathcal{W}}}_{Minimax}(\mathcal{A}_{\Phi^*})$ denote the Minimax score of $\mathcal{A}_{\Phi^*}$ in $\mathcal{C}_{\mathcal{W}}$, i.e., the maximum number of defeating agents in the worst defeat of $\mathcal{A}_{\Phi^*}$ when comparing with other alternatives in $\mathcal{C}_{\mathcal{W}}$. If $r = Smith/Minimax$, the distance between the Minimax score of the chosen decision and the Smith/Minimax winner is bounded by $s^{\mathcal{C}_{\mathcal{W}}}_{Minimax}(\mathcal{A}_{\Phi^*}) - \lfloor \frac{n}{2} \rfloor - 1$.*

Proof Let $\mathcal{S}$ denote the set of Smith members in $\mathcal{W}$, and $\mathcal{C}_{\mathcal{S}}$ the set of alternatives corresponding to the nodes in $\mathcal{S}$, then $\mathcal{C}_{\mathcal{S}} \subseteq \mathcal{C}_{\mathcal{W}}$, $\mathcal{A}_{\Phi^*}$ is a Minimax winner in $\mathcal{C}_{\mathcal{S}}$ and it majority-dominates every candidate in $\mathcal{C}_{\mathcal{W}} \setminus \mathcal{C}_{\mathcal{S}}$. Assume $\mathcal{A}_{\Phi^*}$ is not a Smith/Minimax winner in $\mathcal{X}$. Since $\mathcal{A}_{\Phi^*}$ dominates every outcome in $\mathcal{C}_{\mathcal{L}}$, the Minimax score of $\mathcal{A}_{\Phi^*}$ in $\mathcal{X} = \mathcal{C}_{\mathcal{W}} \cup \mathcal{C}_{\mathcal{L}}$ is then $s^{\mathcal{X}}_{Minimax}(\mathcal{A}_{\Phi^*}) = s^{\mathcal{C}_{\mathcal{W}}}_{Minimax}(\mathcal{A}_{\Phi^*}) = s^{\mathcal{C}_{\mathcal{S}}}_{Minimax}(\mathcal{A}_{\Phi^*})$.

If $\mathcal{A}_{\Phi^*}$ is not a Smith/Minimax winner, then any other outcome $\mathbf{x} \in \mathcal{C}_{\mathcal{S}}$ must not be a Smith/Minimax winner. This is because $s^{\mathcal{C}_{\mathcal{W}}}_{Minimax}(\mathbf{x}) \geq s^{\mathcal{C}_{\mathcal{W}}}_{Minimax}(\mathcal{A}_{\Phi^*})$ and $\mathbf{x}$ might not dominate all the alternatives in $\mathcal{C}_{\mathcal{L}}$.

If $\mathbf{x} \notin \mathcal{C}_{\mathcal{W}}$ is a Smith/Minimax winner, then the best Minimax score of $\mathbf{x}$ is $\lfloor \frac{n}{2} \rfloor + 1$, since it is at least majority dominated by $\mathcal{A}_{\Phi^*}$ and the minimum number of agents to form a majority is $\lfloor \frac{n}{2} \rfloor + 1$. Therefore, the upper bound of the difference between the Minimax score of the chosen candidate $\mathcal{A}_{\Phi^*}$ and the Smith/Minimax winner is $s^{\mathcal{C}_{\mathcal{W}}}_{Minimax}(\mathcal{A}_{\Phi^*}) - \lfloor \frac{n}{2} \rfloor - 1$. □

4 Towards a more efficient process for preferences represented by CP-nets

As we mentioned before, the order σ used to generate the MDTree does not change the fact that the final decision is a Smith member, however, it may affect the size of the search space, and thus, the running time of making a collective decision. Obviously, the fewer outcomes being explored (the smaller the MDTree is), the more efficient the protocol is, as it requires fewer dominance queries from each agent. In general, we do not assume the agents' preference structures are known, and the order σ can be randomly chosen. However, if the agents would like to reveal their preference structures (i.e., only the dependency relations between variables), we can use this information to decide on an order σ that potentially[9] results in a more efficient process of collective decision-making.

Definition 7 (*Minimal common ordered partition*) Let $\mathbf{V}_{\sigma_1} \dots \mathbf{V}_{\sigma_q}$ $(q < m)$ be an ordered partition of $\mathbf{V}$, we say $\mathbf{V}_{\sigma_1} \dots \mathbf{V}_{\sigma_q}$ is a *minimal common* ordered partition for all agents $i \leq n$ if: $\forall \ell < q$, $\mathbf{V}_{\sigma_\ell}$ is a *minimum* subset of $\mathbf{V} - (\mathbf{V}_{\sigma_1} \cup \dots \cup \mathbf{V}_{\sigma_{\ell-1}})$ s.t. $\forall X \in \mathbf{V}_{\sigma_\ell}, (Pa_1(X) \cup \dots \cup Pa_n(X)) \subseteq (\mathbf{V}_{\sigma_1} \cup \dots \cup \mathbf{V}_{\sigma_\ell})$.

Given a minimal common ordered partition $\mathbf{V}_{\sigma_1} \dots \mathbf{V}_{\sigma_q}$, for each agent i $(i \leq n)$, the preference over $\mathbf{V}_{\sigma_\ell}$ $(\ell < q)$ does not depend on any variables in $\mathbf{V}_{\sigma_{\ell+1}} \cup \dots \cup \mathbf{V}_{\sigma_q}$. In such case, the agents may be more willing to compromise on the values of the variables in $\mathbf{V}_{\sigma_k}$ $(k > \ell)$ rather than the values of the variables in $\mathbf{V}_{\sigma_\ell}$. For instance, in an acyclic CP-net, an alternative that violates the values of $Pa(X)$ will never be preferred to another alternative that keeps more preferred values of $Pa(X)$, no matter what value is assigned to X. As a result, an agent will be more willing to compromise on the values of the children variables but keep a preferred value for the parent variables. Therefore, if the variables in $\mathbf{V}_{\sigma_\ell}$ are decided before the variables in $\mathbf{V}_{\sigma_k}$ $(k > \ell)$, an agent may prefer to compromise on the values of the variables in $\mathbf{V}_{\sigma_k}$, rather than compromising on the values of the variables in $\mathbf{V}_{\sigma_\ell}$ by going backtrack to make new proposals on the upper levels of the MDTree. Thus, the number of backtracking might be reduced.

Moreover, although computing the above mentioned aggregation rules in combinatorial domains is hard in general, if the agents' preference structures are more similar it can be easy. Therefore, the smaller the size of each $\mathbf{V}_{\sigma_\ell}$ $(1 \leq \ell \leq q)$ in the minimum

[9] Of course, since the agents' conditional preference tables are unknown, there is no guarantee about which order will be more efficient.

Algorithm 2: `varOrder`

Input: a set of n dependency graph $\mathbf{G} = \{G_1, \ldots, G_n\}$, where each G_i $(i \leq n)$ is a tuple $(\mathbf{V}, E_i)$. $\mathbf{V}$ is the set of nodes in G_i, each represents a variable (in this scenario, all the agents have the same set of variables $\mathbf{V}$); and E_i is a set of directed edges between variables in G_i.

Output: $\sigma = X_{\sigma_1} > \cdots > X_{\sigma_m}$

1 **foreach** $G_i \in \mathbf{G}$ **do**
2 Make G_i a weighted dependency graph and compute the degree of dependency $\mathtt{Dep}_i(X)$ of each variable X in G_i
3 **end**
`// Build a directed graph` G^* `from` $\mathbf{G}$
4 $G^* \leftarrow (\mathbf{V}, E_1 \cup \cdots \cup E_n)$;
5 **foreach** $X \in \mathbf{V}$ **do**
 `// Dep(`X`) is the degree of dependency of` X `in` G^*
6 $\mathtt{Dep}(X) \leftarrow \sum_{i \leq n} \mathtt{Dep}_i(X)$;
7 **end**
8 $\sigma \leftarrow \emptyset$;
9 **repeat**
10 find a maximal mutually connected subset $\mathbf{V}_\sigma$ of G^*; `/* A maximal mutually connected subset of` G^* `is a subset of nodes in` G^* `such that there is a path between any two nodes in the set, but there is no path from a node outside the set to a node inside the set. */`
11 sort variables in $\mathbf{V}_\sigma$ in ascending order according to `Dep` in G^*;
12 **append** all variables in $\mathbf{V}_\sigma$ **to** σ;
13 remove $\mathbf{V}_\sigma$ from G^* and all edges connecting to G^*.
14 **until** $G^* = \emptyset$;
15 **return** σ;

common ordered partition is, the more likely that they explore fewer alternatives. For instance, suppose that all the agents' preferences are represented by CP-nets and their CP-nets are compatible with a linear order $\sigma = X_{\sigma_1} > \ldots X_{\sigma_m}$[10]. Then, $\forall 1 \leq \ell \leq q$, $|\mathbf{V}_{\sigma_\ell}| = 1$. In this case, for each agent, every variable X_{σ_i} is preferentially independent of $X_{\sigma_{i+1}}, \ldots, X_{\sigma_m}$. Therefore, if we generate the MDTree following σ, once the majority preference of variables $X_{\sigma_1}, \ldots, X_{\sigma_i}$ is determined, i.e., the open nodes which represent the partial assignments to the varaibles $X_{\sigma_1}, \ldots, X_{\sigma_i}$ are created, the agents will only follow these branches (i.e., backtracking-free). Other possible assignments to the variables $X_{\sigma_1}, \ldots, X_{\sigma_i}$ will be pruned out. As such, the `MDTSC` protocol is similar to an issue-by-issue sequential voting procedure, under which computing a decision is easy.

Algorithm 2 provides a heuristic technique to compute an order of variables based on the agents preference structures (dependency graphs). On a high level, it computes a minimal common ordered partition $\mathbf{V}_{\sigma_1} \ldots \mathbf{V}_{\sigma_q}$ of $\mathbf{V}$ and orders the variables accordingly (lines 9–14; Algorithm 2). Moreover, because the maximal size of $\mathbf{V}_{\sigma_\ell}$ $(1 \leq \ell \leq m)$ in general could be m, Algorithm 2 also provides a heuristic to rank the variables inside each element of the minimal common ordered partition. This heuristic is based on the degree of dependency (`Dep`) of the variables, where $\mathtt{Dep}(X)$ of a vari-

[10] A CP-net $\mathcal{N}$ is said to be compatible with a linear order $\sigma = X_{\sigma_1} > \ldots X_{\sigma_m}$ if the dependency graph $\mathcal{G}_\mathcal{N}$ is compatible with σ, i.e., $\forall X, Y \in \mathbf{V}$ there is an edge (X, Y) in $\mathcal{G}_\mathcal{N}$ only if $X > Y$ in σ.

able X is the sum of the degrees of dependency $\texttt{Dep}_i(X)$ in each agent's preference network G_i, $i \leq n$ (lines 5–7; Algorithm 2). In an agent's preference network, the degree of dependency of a variable is defined as follows:

Definition 8 Given a dependency graph $G_i = (\mathbf{V}, E_i)$, the *weighted dependency graph* assigns a weight to every arc in $\langle X, Y\rangle$ in E_i top-down, such that the weight $w_i(\langle X, Y\rangle)$ is equal to the sum of the weighed indegrees of X plus 1: $w_i(\langle X, Y\rangle) \leftarrow \sum_{\langle Z,X\rangle \in E_i} w_i(\langle Z, X\rangle) + 1$. An arc origining from a variable without any parent has the weight of 1. The *degree of dependency* of a variable X in G_i, denoted by $Dep_i(X)$, is then the sum of the weighted indegrees of X.

5 Handling nonstrict and incomplete preferences

In the previous section, we present our proposed protocol assuming that the inputs are complete and strict profiles. In this section, we are concerned with the cases where the agents' preferences are incomplete and/or nonstrict preferences. As we mentioned before, the best possible alternative (BPA) of a node in the MDTree is unique when an agent's preference is complete and strict. However, when it is not the case, there may exist more than one BPA of a single node, which are indifferent to or incomparable to each other. Therefore, the best possible alternative of a node Φ for an agent i ($\texttt{BPA}_i(\Phi)$) is no longer a singleton; instead, it becomes a set. We therefore generalize $\texttt{BPA}_i(\Phi)$ as follows:

Definition 9 (*Best possible alternative (generalized)*) At each node $\Phi \in \mathcal{L}_i$, the best possible alternative (BPA) of agent i, denoted by $\texttt{BPA}_i(\Phi)$, is a subset of completions of $\mathcal{A}_\Phi$ ($\texttt{BPA}_i(\Phi) \subseteq \texttt{Comp}(\mathcal{A}_\Phi)$) such that every alternative in $\texttt{BPA}_i(\Phi)$ is not less preferred to any completion of $\mathcal{A}_\Phi$: $\texttt{BPA}_i(\Phi) = \{\mathbf{x} \in \texttt{Comp}(\mathcal{A}_\Phi) \mid \forall \mathbf{y} \in \texttt{Comp}(\mathcal{A}_\Phi), \mathbf{y} \not\succ_i \mathbf{x}\}$.

Proposition 2 *At each node $\Phi \in \mathcal{L}_i$, for all $\mathbf{y} \in \texttt{Comp}(\mathcal{A}_\Phi)$, if $\mathbf{y} \notin \texttt{BPA}_i(\Phi)$, then $\exists \mathbf{x} \in \texttt{BPA}_i(\Phi)$, s.t. $\mathbf{x} \succ_i \mathbf{y}$.*

Proof Let $\mathcal{C}_0 = \texttt{Comp}(\mathcal{A}_\Phi) \setminus \texttt{BPA}_i(\Phi)$, assume by way of contradiction that there exists an alternative, denoted by $\mathbf{y}_0$, such that $\mathbf{y}_0 \in \mathcal{C}_0$ and $\nexists \mathbf{x} \in \texttt{BPA}_i(\Phi),\ \mathbf{x} \succ_i \mathbf{y}_0$.

Let q denote the cardinality of $\mathcal{C}_0$, i.e., $q = |\mathcal{C}_0|$, and $\mathcal{C}_1 = \mathcal{C}_0 \setminus \{\mathbf{y}_0\}$ ($|\mathcal{C}_1| = q - 1$). Since $\mathbf{y}_0 \in \mathcal{C}_0$ and $\nexists \mathbf{x} \in \texttt{BPA}_i(\Phi),\ \mathbf{x} \succ_i \mathbf{y}_0$, There must exist another alternative $\mathbf{y}_1 \in \mathcal{C}_1$, such that $\mathbf{y}_1 \succ_i \mathbf{y}_0$. Moreover, $\nexists \mathbf{x} \in \texttt{BPA}_i(\Phi),\ \mathbf{x} \succ_i \mathbf{y}_1$; otherwise, $\mathbf{x} \succ_i \mathbf{y}_1 \succ_i \mathbf{y}_0$, contradicting our assumption that $\forall \mathbf{x} \in \texttt{BPA}_i(\Phi),\ \mathbf{x} \not\succ_i \mathbf{y}_0$.

By inductively applying the same argument as mentioned above, we can construct a chain of alternatives $\{\mathbf{y}_0, \ldots, \mathbf{y}_k\}$, such that $\mathbf{y}_t \succ_i \mathbf{y}_{t-1}$ for $t = 1, \ldots, k$, and $\nexists \mathbf{y} \in \mathcal{C}_0$ such that $\mathbf{y} \succ_i \mathbf{y}_k$. Clearly, $k < q$. Since $\mathcal{C}_0$ is of limited size, there must exists $\mathbf{x}_0 \in \texttt{BPA}_i(\Phi)$, such that $\mathbf{x}_0 \succ_i \mathbf{y}_k$; otherwise, $\mathbf{y}_k \in \texttt{BPA}_i(\Phi)$, contradicting the assumption that $\mathbf{y}_k \in \mathcal{C}_0$. Therefore, $\mathbf{x}_0 \succ_i \mathbf{y}_k \succ_i \cdots \succ_i \mathbf{y}_0$. However, this contradicts with our assumption that $\forall \mathbf{x} \in \texttt{BPA}_i(\Phi),\ \mathbf{x} \not\succ_i \mathbf{y}_0$.

Therefore, the assumption $\exists \mathbf{y}_0 \in \texttt{Comp}(\mathcal{A}_\Phi) \setminus \texttt{BPA}_i(\Phi)$ s.t. $\forall \mathbf{x} \in \texttt{BPA}_i(\Phi),\ \mathbf{x} \not\succ_i \mathbf{y}_0$ will not hold. Hence, $\forall \mathbf{y} \in \texttt{Comp}(\mathcal{A}_\Phi) \setminus \texttt{BPA}_i(\Phi), \exists \mathbf{x} \in \texttt{BPA}_i(\Phi)$ such that $\mathbf{x} \succ_i \mathbf{y}$. □

Note that $\forall \mathbf{x}, \mathbf{x}' \in \mathtt{BPA}_i(\Phi)$, either $\mathbf{x} \sim_i \mathbf{x}'$ or $\mathbf{x} \bowtie_i \mathbf{x}'$ hold. Moreover, it could also be the case that an outcome in the BPA of a node is indifferent or incomparable to an outcome in the BPA of another node (this can happen even when each node has a unique outcome in the BPA). In order to handle incompleteness and indifference when comparing leaf nodes of an MDTree, we generalize the notion of preference relations over leaf nodes as follows, based on a collective evaluation over the BPAs:

Definition 10 (*Preference relations between leaf nodes (generalized)*) Given an agent i and a pair of leaf nodes Φ and Φ' ($\Phi, \Phi' \in \mathcal{L}_i$), we say:

- Φ is at least as preferred as Φ' for agent i (written as $\Phi \succeq_i^+ \Phi'$) iff $\forall \mathbf{x}' \in \mathtt{BPA}_i(\Phi')$, $\exists \mathbf{x} \in \mathtt{BPA}_i(\Phi)$, such that $\mathbf{x} \succeq_i \mathbf{x}'$;
- Φ is strictly preferred to Φ' for agent i (written as $\Phi \succ_i^+ \Phi'$) iff $\Phi \succeq_i^+ \Phi'$ and $\Phi' \not\succeq_i^+ \Phi$;
- Φ is indifferent to Φ' for agent i (written as $\Phi \sim_i^+ \Phi'$) iff $\Phi \succeq_i^+ \Phi'$ and $\Phi' \succeq_i^+ \Phi$;
- Φ is incomparable to Φ' for agent i (written as $\Phi \bowtie_i^+ \Phi'$) iff $\Phi \not\succeq_i^+ \Phi'$ and $\Phi' \not\succeq_i^+ \Phi$;

Lemma 1 *For each agent i, $\succeq_i^+$ is reflexive and transitive.*

Proof We first prove the reflexibility of $\succeq_i^+$. For any $\Phi \in \mathcal{L}_i$, as individual preference $\succeq_i$ over outcomes are reflexive, $\forall \mathbf{x} \in \mathtt{BPA}_i(\Phi)$, $\mathbf{x} \succeq_i \mathbf{x}$. Therefore, $\forall \mathbf{x} \in \mathtt{BPA}_i(\Phi)$, $\exists \mathbf{x}' \in \mathtt{BPA}_i(\Phi)$ and $\mathbf{x}' = \mathbf{x}$, such that $\mathbf{x} \succeq \mathbf{x}'$. Hence, $\Phi \succeq_i^+ \Phi$ and $\succeq_i^+$ is reflexive.

Regarding transitivity, assume $\Phi, \Phi', \Phi'' \in \mathcal{L}_i$, $\Phi \succeq_i^+ \Phi'$ and $\Phi' \succeq_i^+ \Phi''$. $\forall \mathbf{x}'' \in \mathtt{BPA}_i(\Phi'')$, $\exists \mathbf{x}' \in \mathtt{BPA}_i(\Phi')$ such that $\mathbf{x}' \succeq_i \mathbf{x}''$. As $\Phi \succeq_i^+ \Phi'$, $\exists \mathbf{x} \in \mathtt{BPA}_i(\Phi)$ such that $\mathbf{x} \succeq_i \mathbf{x}'$. Since individual preference $\succeq_i$ over the outcomes are transitive, $\mathbf{x} \succeq_i \mathbf{x}' \succeq_i \mathbf{x}''$. Therefore, $\forall \mathbf{x}'' \in \mathtt{BPA}_i(\Phi'')$, $\exists \mathbf{x} \in \mathtt{BPA}_i(\Phi)$ such that $\mathbf{x} \succeq_i \mathbf{x}''$. $\Phi \succeq_i^+ \Phi''$ and $\succeq_i^+$ is transitive. □

By Definition 10, each individual agent i can then define a partial order over the set of leaf nodes $\mathcal{L}_i$. As such, he can still maintain a priority queue of leaf nodes based on its preference relation $\succeq_i^+$ over $\mathcal{L}_i$. Given Φ^*, an agent will not stop proposing if there exist at least one leaf node Φ, such that $\exists \mathbf{x} \in \mathtt{BPA}_i(\Phi)$ such that $\mathbf{x} \succ_i \mathcal{A}_{\Phi^*}$. Moreover, he will choose the optimal one among all the feasible leaf nodes. Therefore, the BATCD strategy can be modified to deal with incomplete and nonstrict preferences:

Definition 11 (*BATCD (Generalized)*) Given a list of feasible leaf nodes $\mathcal{L}_i$ and Φ^* in the current iteration, agent i ($i \leq n$) will choose one of the leaf nodes $\Phi \in \mathcal{L}_i$ to propose on, such that:

- $\forall \Phi' \in \mathcal{L}_i$, $\Phi' \not\succ_i^+ \Phi$; and
- $\Phi^* = NULL$ or $\exists \mathbf{x} \in \mathtt{BPA}_i(\Phi)$ such that $\mathbf{x} \succ_i \mathcal{A}_{\Phi^*}$.

If there exist no such leaf nodes, agent i will then stop making proposals and stick with the reference node Φ^*. Moreover, in general, when there is more than one leaf node satisfying the above condition, we assume that agent i will give higher priority to a node Φ than another node Φ' if the depth of Φ is greater than that of Φ', i.e., $|\mathcal{A}_\Phi| > |\mathcal{A}_{\Phi'}|$. If there is more than one optimal node at the same depth of the MDTree,

agent i can randomly choose one to propose. In the following, we prove that when we allow incomplete and nonstrict preferences, the final decision made by the MDTSC protocol is still guaranteed to be a Smith member, following the definition of Smith set in Brandt et al. (2009).

Proposition 3 *For an individual agent i ($i \leq n$), given a* winning *node Φ^* and a node Φ (not necessarily at depth m), if $\Phi \not\succ_i^+ \Phi^*$, then: $\forall \mathbf{y} \in \mathtt{Comp}(\mathcal{A}_\Phi)$, $\mathbf{y} \not\succ_i \mathcal{A}_{\Phi^*}$.*

Proof According to Definition 10, if $\Phi \not\succ_i^+ \Phi^*$, then either $\Phi^* \succeq_i^+ \Phi$ or $\Phi \not\succeq_i^+ \Phi^*$ holds. If $\Phi^* \succeq_i^+ \Phi$, then $\forall \mathbf{y} \in \mathtt{Comp}(\mathcal{A}_\Phi)$, $\mathcal{A}_{\Phi^*} \succeq_i \mathbf{y}$, and so $\mathbf{y} \not\succ_i \mathcal{A}_{\Phi^*}$. Or if $\Phi \not\succeq_i^+ \Phi^*$, then $\nexists \mathbf{x} \in \mathtt{BPA}_i(\Phi)$, such that $\mathbf{x} \succeq_i \mathcal{A}_{\Phi^*}$. Therefore, $\forall \mathbf{x} \in \mathtt{BPA}_i(\Phi)$, $\mathbf{x} \not\succeq_i \mathcal{A}_{\Phi^*}$.

Now, assume by contradiction that there exists an alternative $\mathbf{y} \in \mathtt{Comp}(\mathcal{A}_\Phi) \setminus \mathtt{BPA}_i(\Phi)$ such that $\mathbf{y} \succ_i \mathcal{A}_{\Phi^*}$. According to Proposition 2, $\exists \mathbf{x} \in \mathtt{BPA}_i(\Phi)$ such that $\mathbf{x} \succ_i \mathbf{y}$. As individual preference is transitive, then we have $\mathbf{x} \succ_i \mathbf{y} \succ_i \mathcal{A}_{\Phi^*}$, contracting the conclusion we draw before: $\forall \mathbf{x} \in \mathtt{BPA}_i(\Phi)$, $\mathbf{x} \not\succeq_i \mathcal{A}_{\Phi^*}$. Therefore, $\forall \mathbf{y} \in \mathtt{Comp}(\mathcal{A}_\Phi)$, $\mathbf{y} \not\succ_i \mathcal{A}_{\Phi^*}$. □

Let $\#\{i \leq n : \Phi \succ_i^+ \Phi'\}$ denote the number of agents who prefer Φ over Φ', in the following section we say a node Φ majority dominates another node Φ', written as $\Phi \succ_{maj}^+ \Phi'$, if $\#\{i \leq n : \Phi \succ_i^+ \Phi'\} > \frac{n}{2}$; otherwise, $\Phi \not\succ_{maj}^+ \Phi'$.

Corollary 2 *Given a* winning *node Φ^* and a node Φ (not necessarily at depth m), if $\Phi \not\succ_{maj}^+ \Phi^*$, then: $\forall \mathbf{y} \in \mathtt{Comp}(\mathcal{A}_\Phi)$, $\mathbf{y} \not\succ_{maj} \mathcal{A}_{\Phi^*}$.*

Let $\mathcal{L}$ denote the union of $\mathcal{L}_i$, $i \leq n$ ($\mathcal{L} = \bigcup_{i \leq n} \mathcal{L}_i$); $\mathcal{C}_\mathcal{W}$ denote the outcomes specified by the nodes in $\mathcal{W}$, i.e., $\mathcal{C}_\mathcal{W} = \{\mathcal{A}_\Phi | \Phi \in \mathcal{W}\}$; and $\mathcal{C}_\mathcal{L}$ denote all possible completions to the partial assignments specified by the nodes in $\mathcal{L}$, i.e., $\mathcal{C}_\mathcal{L} = \bigcup_{\Phi \in \mathcal{L}} \mathtt{Comp}(\mathcal{A}_\Phi)$. Now, we prove that the proposed protocol always chooses a Smith member without assuming that the agents' preferences are complete and strict.

Theorem 4 *If Φ^* is a Smith member in $\mathcal{W}$, then the final collective decision $\mathcal{A}_{\Phi^*}$ chosen by MDTSC protocol is guaranteed to be a member of the Smith set.*

Proof Let $\mathcal{S}$ be the Smith set in $\mathcal{W}$, and $\mathcal{C}_\mathcal{S}$ be the outcomes corresponding to the nodes in $\mathcal{S}$. As $\mathcal{A}_{\Phi^*}$ is a Smith member in $\mathcal{C}_\mathcal{W}$, therefore $\mathcal{A}_{\Phi^*} \in \mathcal{C}_\mathcal{S}$. $\forall \mathbf{x} \in \mathcal{C}_\mathcal{S}$ and $\mathbf{x} \neq \mathcal{A}_{\Phi^*}$, either there is a majority preference cycle between $\mathcal{A}_{\Phi^*}$ and $\mathbf{x}$; or there is a majority tie or incomparability between $\mathcal{A}_{\Phi^*}$ and $\mathbf{x}$ ($\mathbf{x} \not\succ_{maj} \mathcal{A}_{\Phi^*}$ and $\mathcal{A}_{\Phi^*} \not\succ_{maj} \mathbf{x}$). In any of the above cases, if $\mathbf{x} \in \mathcal{C}_\mathcal{W} \setminus \{\mathcal{A}_{\Phi^*}\}$ happens to be a Smith member in $\mathcal{X}$, then $\mathcal{A}_{\Phi^*}$ must also be a Smith member in $\mathcal{X}$.

As $\mathcal{C}_\mathcal{S}$ is the Smith set in $\mathcal{C}_\mathcal{W}$, $\forall \mathbf{x} \in \mathcal{C}_\mathcal{W} \setminus \mathcal{C}_\mathcal{S}$, $\mathcal{A}_{\Phi^*} \succ_{maj} \mathbf{x}$. Hence, if $\mathbf{x} \in \mathcal{C}_\mathcal{W} \setminus \mathcal{C}_\mathcal{S}$ happens to be a Smith member in $\mathcal{X}$, then $\mathcal{A}_{\Phi^*}$ must also be a Smith member in $\mathcal{X}$.

When all the agents stop making proposals and stick with Φ^*, then for every agent i, $\forall \Phi \in \mathbf{L}_i$, $\nexists \mathbf{x} \in \mathtt{BPA}_i(\Phi)$ such that $\mathbf{x} \succ_i \mathcal{A}_{\Phi^*}$, and so $\Phi \not\succ_i^+ \Phi^*$. Moreover, $\forall \Phi \in \mathbf{L}$, for a majority of agents, $\nexists \mathbf{x} \in \mathtt{BPA}_i(\Phi)$ such that $\mathbf{x} \succ_i \mathcal{A}_{\Phi^*}$, because otherwise, Φ

will receive a majority of agents' proposals and will continue to be explored. Hence, $\Phi \not\succ^{+}_{maj} \Phi^{*}$.

According to Corollary 2, $\forall \Phi \in \mathbf{L}, \forall \mathbf{y} \in \texttt{Comp}(\mathcal{A}_{\Phi}), \mathbf{x} \not\succ_{maj} \mathcal{A}_{\Phi^{*}}$. Therefore, if $\mathbf{x} \in \mathcal{X} \setminus \mathcal{C}_{\mathcal{W}}$ happens to be a Smith member in $\mathcal{X}$, then $\mathcal{A}_{\Phi^{*}}$ must also be a Smith member in $\mathcal{X}$.

Finally, from the above analysis we can conclude that $\forall \mathbf{x} \in \mathcal{X} \setminus \{\mathcal{A}_{\Phi^{*}}\}$, if $\mathbf{x}$ happens to be a Smith member in $\mathcal{X}$, then $\mathcal{A}_{\Phi^{*}}$ must also be a Smith member in $\mathcal{X}$. Hence $\mathcal{A}_{\Phi^{*}}$ must be a Smith member in $\mathcal{X}$. □

6 Experiments

We now describe some results of experiments with respect to (i) the number of variables and the size of the alternative space $|\mathcal{X}|$; (ii) the average number of alternatives that each agent explores during the decision-making process, denoted by Alt_{con};[11] (iii) the average number of dominance queries (outcome comparisons) that each agent needs to answer, denoted by DQ; (iv) the average number of alternatives that requires direct pairwise comparisons over, i.e., the average size of $\mathcal{W}$ ($|\mathcal{W}|$); and (v) the average running time. We test the performance of the protocol when using a random order as well as when using an order computed by Algorithm 2. The results of using a random order is denoted by `MDTSC`, and the results of using an order computed by Algorithm 2 is denoted by `MDTSC*`. Moreover, we also compare the running time of the proposed protocol with an algorithm, called `SMTarjan`. `SMTarjan` algorithm first computes the Smith set using *Tarjan's strongly connected components algorithm* (Tarjan 1972) from a weighted majority graph of the preference profile. Then, it chooses a Minimax winner in the Smith Set based on the maximum weight of the incoming edges of the vertices.

In these experiments, we focus on binary variables. We only present the results with the Smith/Minimax rule. The experimental results of other Condorcet-consistent rules are similar to those with Smith/Minimax rule. For the sake of simplicity, we consider the representation language "SLO SCPnet" (Domshlak et al. 2006) to represent the agents' preferences in combinatorial domains. Each SLO SCPnet induces a strict linear order over the alternative space. In those experiments, we first randomly generate a CP-net and then convert it to a SLO SCPnet. The dependency graphs of the each CP-net is acyclic. SLO SCPnet is a compact representation language that combines the nature of soft constraints and *ceteris paribus* structural preferences. Moreover, as outcome comparisons in SLO SCPnet is easy, the experiment results reflect the reduction of the total number of dominance queries using the proposed protocol compared to using `SMTarjan` algorithm. Obviously, when the dominance queries in the preference representation are more difficult (e.g., dominance queries in CP-nets), this saving will be even bigger, since longer time is needed for answering each dominance query.

[11] Notice that Alt_{con} is the number of different outcomes each agent encounters when she generates the MDTree. In essence, it is equal to the number of leaf nodes on the MDTree at the end of the decision-making process.

6.1 General results with no required common preference structures

In the first set of experiments, there are 15 agents; each represents his preference by a SLO SCPnet, and there is no required common structure of the preferences between different agents. When the number of variables is smaller than 8, we run 5,000 rounds of experiments. Subsequently, we run 500 rounds of experiments for each number of variables that is larger than 8 (variables 10, 12, 14 and 16). In each round of experiment, we randomly generate 15 different SLO SCP-nets (representing different agent's preferences in the group) with the given variable number. We then execute the proposed protocol, and get an average result over all the test rounds as the result of the given variable number. However, when the number of variables is larger than 10, we do not test the running time of the `SMTarjan` algorithm, since a single run exceeds 10 h.

From Table 1, we can see that the average number of outcomes Alt_{con} that each agent need to consider, the average number of dominance queries DQ, the average size of $|\mathcal{W}|$, and average the running time of the `MDTSC` protocol are increasing as the number of variables increases. Nonetheless, making a collective decision using the `MDTSC` protocol is more efficient than using `SMTarjan` algorithm. When the number of variables is large, the `MDTSC` protocol offers several orders of magnitude reduction in running time compared to the `SMTarjan` algorithm. Moreover, compared to the large outcome space, using the `MDTSC` protocol, the average number of alternatives Alt_{con} that each agent needs to consider is significantly reduced. When the number of variables is large, e.g. 16, on average, the size of $|\mathcal{W}|$ is around 1,000, which is less than one fiftieth of the entire alternative space.

Comparing the performance of `MDTSC` and `MDTSC*`, we can see that although there is no required common preference structures between the agents, using the variable order generated by Algorithm 2 still offers considerable save on the average number of considered outcome Alt_{con}, the average number of dominance testing DQ, as well as the average running time in general.

Moreover, in these experiments, we observe that the winner in $\mathcal{C}_{\mathcal{W}}$ is often identical to the winner in $\mathcal{X}$ $(= \mathcal{C}_{\mathcal{W}} \cup \mathcal{C}_{\mathcal{L}})$. For each number of variables, in more than 99.9 % cases the final decision chosen by `MDTSC` or `MDTSC*` protocol is indeed, the Smith/Minimax candidate. Moreover, even in the cases that the proposed protocol fails to choose a Smith/Minimax candidate, the Minimax score of the alternative chosen by `MDTSC` or `MDTSC*` is sufficiently close to that of a Smith/Minimax candidate (the difference between the Minimax scores on average is <1.5).

6.2 Results with similar preference structures

Subsequently, we test the average reduction in computation using the variable order computed by Algorithm 2, compared to that using a random order to generate the MDTree. We also investigate the situations with different similarities among the agents' preference structures. We generate the agents' preference networks with different restrictions on the maximum sizes of the elements in the minimal common ordered partition, denoted by $Max_{|\mathbf{V}_{\sigma_\ell}|}$. In these experiments, $Max_{|\mathbf{V}_{\sigma_\ell}|}$ is restricted

Table 1 Results with no required common preference structures between agents (15 agents, 2–16 variables)

Var.	$\lvert\mathcal{X}\rvert$	Alt_{con}		DQ		$\lvert\mathcal{W}\rvert$		Time (s)		
		MDTSC	MDTSC*	MDTSC	MDTSC*	MDTSC	MDTSC*	SMTarjan	MDTSC	MDTSC*
2	4	3.5	2.24	1.8	1.37	1.6	1.35	0.015	0.009	0.0072
4	16	13.0	8.97	34.9	15.67	5.6	3.92	0.49	0.052	0.038
6	32	44.2	25.75	400.9	137.36	15.5	11.11	11.9	0.25	0.20
8	256	147.8	76.44	4176.7	1161.15	45.3	36.14	254.2	1.39	0.81
10	1024	459.4	222.9	42739.3	9218.66	129.8	122.62	5025.7	8.51	3.28
12	4096	1479.0	659.84	466050.	69741.	398.9	395.54	NA	75.1	40.84
14	16384	3167.2	1605.205	3.8×10^6	947469.	800.8	783.56	NA	350.1	210.63
16	65536	4399.0	2011.71	6.2×10^6	1.58×10^6	1045.2	969.714	NA	855.5	506.35

Table 2 Performance comparison when the agents have similar preference structures (15 agents, 10 variables)

$Max_{\|\mathbf{V}_{\sigma_\ell}\|}$	Alt_{con}		DQ		$\|\mathcal{W}\|$		Time (s)	
	MDTSC	MDTSC*	MDTSC	MDTSC*	MDTSC	MDTSC*	MDTSC	MDTSC*
1	349.55	11.	36664.8	5.00	61.23	1.	7.39	0.09
3	448.83	128.12	41955.1	3428.15	124.37	92.33	7.96	2.19
5	450.63	170.48	41947.7	4751.33	120.02	113.60	8.36	2.67
7	456.55	188.07	43301.5	5786.84	122.26	116.39	8.32	2.80
9	461.20	207.07	45188.3	7580.51	129.34	117.77	8.54	2.98
10	459.4	222.90	42739.3	9218.66	129.8	122.62	8.51	3.28

to 1–10 respectively, and there are 15 agents and 10 variables. Table 2 demonstrates the results using the order computed by Algorithm 2. Clearly, when $Max_{|\mathbf{V}_{\sigma_\ell}|}$ is smaller, on average, making a collective decision is easier: the average numbers of alternatives considered and dominance queries are fewer, and the running time is shorter.

Table 2 also provides a comparison to the proposed protocol when using a random variable order. Comparing the results of MDTSC to that of MDTSC*, we can see that on average, using Algorithm 2 in the proposed protocol is more efficient: the agents answer fewer dominance queries and explore fewer alternatives during decision-making.

7 Conclusion

In this paper, we proposed a general distributed protocol for collective decision-making in combinatorial domains. The proposed protocol satisfies the Smith criterion and the final outcome chosen is often identical to the winner of the given rule. It works under incomplete information and significantly reduces the search space.

An important issue for future research is manipulation: how an agent will get a better alternative by not following the BATCD strategy, how can we detect those untruthful behaviors, and how to avoid them. Moreover, although the protocol can be applied to most preference representation languages, the running time of the protocol strongly depends on the running time of answering dominance queries and optimization queries. Thus, it is important to investigate the feasibility of the proposed protocol when applied to different representation languages with different computational properties on preference queries.

References

Arrow, K.J., Sen, S.K., Suzumura, K. (eds.): Handbook of Social Choice and Welfare, vol. 1. Elsevier, Amsterdam (2002)

Arrow, K.J.: Social Choice and Individual Values. Wiley, New York (1951)

Bacchus, F., Adam, G.: Graphical models for preference and utility. In: Proceedings of the Proceedings of the Eleventh Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-95), pp. 3–10. Morgan Kaufmann, San Francisco (1995)

Boutilier, C., Brafman, I., Domshlak, C., Hoos, H.H., Poole, D.: Cp-nets: a tool for representing and reasoning with conditional ceteris paribus preference statements. J. Artif. Int. Res. **21**, 135–191 (2004)

Boutilier, C., Hoos, HH.: Bidding languages for combinatorial auctions. In: Proceeding of the 17th International Joint Conference on Artifical, Intelligence (2001)

Bouveret, S., Endriss, U., Lang, J.: Conditional importance networks: a graphical language for representing ordinal, monotonic preferences over sets of goods. In: IJCAI, pp 67–72 (2009)

Brams, S.J., Kilgour, D.M., Zwicker, W.S.: The paradox of multiple elections. Soc. Choice Welf. **15**, 211–236 (1998)

Brandt, F., Fischer, F.A., Harrenstein, P.: The computational complexity of choice sets. Math. Log. Q. **55**(4), 444–459 (2009)

Cave, J., Salant, S.W.: Cartel quotas under majority rule. Am. Econ. Rev. **85**(1), 82–102 (1995)

Chou, F., Tan, J.: A majority voting scheme in wireless sensor networks for detecting suspicious node. In: Second International Symposium on Electronic Commerce and Security, 2009, ISECS '09. vol. 2, pp. 495–498 (2009)

Domshlak, C., Prestwich, S.D., Rossi, F., Venable, K.B., Walsh, T.: Hard and soft constraints for reasoning about qualitative conditional preferences. J. Heuristics **12**(4), 263–285 (2006)

Epple, D., Romer, T., Sieg, H.: Interjurisdictional sorting and majority rule: an empirical analysis. Econometrica **69**(6), 1437–1465 (2001)

Goldsmith, J., Lang, J., Truszczynski, M., Wilson, N.: The computational complexity of dominance and consistency in CP-nets. J. Artif. Int. Res. **33**(1), 403–432 (2008)

Konieczny, S., Lang, J., Marquis, P.: DA2 merging operators. Artif. Intell. **157**(1–2), 49–79 (2004)

Konieczny, S., Pérez, R.P.: Propositional belief base merging or how to merge beliefs/goals coming from several sources and some links with social choice theory. Eur. J. Oper. Res **160**(3), 785–802 (2005)

Mura, LP., Shoham, Y.: Expected utility networks. In: Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence, UAI'99, pp. 366–373. Morgan Kaufmann Publishers Inc., San Francisco (1999)

Lang J.: From preference representation to combinatorial vote. In: KR, pp. 277–290 (2002)

Lang, J.: Logical preference representation and combinatorial vote. Ann. Math. Artif. Intell. **42**, 37–71 (2004)

Lang, J., Pini, M.S., Rossi, F., Salvagnin, D., Venable, K.B., Walsh, T.: Winner determination in voting trees with incomplete preferences and weighted votes. Auton. Agent. Multi-Agent Syst. **25**(1), 130–157 (2012)

Lang, J., Xia, L.: Sequential composition of voting rules in multi-issue domains. Math. Soc. Sci. **57**(3), 304–324 (2009)

Li, M., Vo, Q.B., Kowalczyk, R.: An efficient protocol for negotiation over combinatorial domains with incomplete information. In: Proceedings of the 27th Conference in Uncertainty in Artificial Intelligence, vol. 2, Barcelona (2011a)

Li, M., Vo, B.Q., Ryszard, K.: Majority-rule-based preference aggregation on multi-attribute domains with CP-nets. In: Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems, vol. 1, pp. 659–666. International Foundation for Autonomous Agents and Multiagent Systems, Richland (2011b)

Schulze, M.: A new monotonic, clone-independent, reversal symmetric, and Condorcet-consistent single-winner election method. Soc. Choice Welf. **36**(2), 267–303 (2011)

Tarjan, R.: Depth-first search and linear graph algorithms. SIAM J. Comput. **1**(2), 146–160 (1972)

Ward, B.: Majority rule and allocation. J. Confl. Resolut. **5**, 379–389 (1961)

Xia, L., Conitzer, V.: Determining possible and necessary winners given partial orders. J. Artif. Intell. Res. **41**, 25–67 (2011)

Xia, L., Conitzer, V., Lang, J.: Voting on multiattribute domains with cyclic preferential dependencies. In: AAAI'08: Proceedings of the 23rd national conference on Artificial intelligence, pp. 202–207. AAAI Press (2008)

Xia, L., Lang, J., Ying M.: Strongly decomposable voting rules on multiattribute domains. In AAAI'07: Proceedings of the 22nd national conference on Artificial intelligence, pp. 776–781. AAAI Press (2007)